

The Spring Framework: **Core Capabilities Part 3**

Originals of Slides and Source Code for Examples: http://courses.coreservlets.com/Course-Materials/spring.html

Customized Java EE Training: http://courses.coreservlets.com/ Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6. Developed and taught by well-known author and developer. At public venues or onsite at your location.





© 2008 coreservlets.com

For live Spring & Hibernate training, see courses at http://courses.coreservlets.com/.

Taught by the experts that brought you this tutorial. Available at public venues, or customized versions can be held on-site at your organization.

- Courses developed and taught by Marty Hall
- Java 5, Java 6, intermediate/beginning servlets/JSP, advanced servlets/JSP, Struts, JSF, Ajax, GWT, custom mix of topics Courses developed and taught by coreservlets.com experts (edited by Marty)

 - Spring, Hibernate/JPA, EJB3, Ruby/Rails

Contact hall@coreservlets.com for details

Topics in This Section

- Referencing other beans
- Plain values
- Properties integration
- Resource integration
- Collection mapping

Java EE training: http://courses.coreservlets.com

© 2008 coreservlets.com



Referencing Other Beans

Customized Java EE Training: http://courses.coreservlets.com/ Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Bean Collaborator Reference

- Collaborator
 - Peer bean in a bean interaction
- Configuration
 - Reference other beans within a common Spring IoC container
 - Define collaborators using local, general or inner bean references

6

lava EE training: http://courses.coreservlets.com

Inner Bean Reference

- Fulfills a single dependency injection instance
- Analogous to anonymous classes

Inner Bean Process

- Nested for XML constructor or property setter DI
 - XML element type bean
 - XML property or constructor-arg child
- Nested reference for XML collection element
 - For example, XML **list** child
- Do not specify identifiers or scope

3

Java EE training: http://courses.coreservlets.com

Inner Bean Example

```
<bean id="bookReader" class="coreservlets.BookReader">
  property name="bookLibrary">
    <bean class="coreservlets.JavaBookLibrary"/>
  </property>
</bean>
<bean id="bookReader" class="coreservlets.BookReader">
  <constructor-arg>
    <bean class="coreservlets.JavaBookLibrary"/>
  </constructor-arg >
</bean>
<br/>
<br/>
dean ...>
  <constructor-arg>
    t>
      <bean class="coreservlets.JavaBookLibrary"/>
      <bean class="coreservlets.JavaBookLibrary"/>
      <bean class="coreservlets.JavaBookLibrary"/>
  </constructor-arg >
</bean>
                                          Java EE training: http://courses.coreservlets.com
```

General Collaborator Reference

- Reference other beans in the Spring IoC container
- Favor as the default method for specifying collaborators

10

General Collaborator Process

- Nested reference for XML constructor or property setter DI
 - XML element type **ref** and attribute **bean**
 - XML property or constructor-arg child
- Short-hand reference for XML constructor
 DI
 - Declare as XML attribute type ref
- Nested reference for XML collection element
 - For example, XML **list** child

General Collaborator Example

```
<bean id="bookReader" class="coreservlets.BookReader">
  <constructor-arg>
    <ref bean="bookLibrary"/>
  </constructor-arg >
</bean>
<bean id="bookReader" class="coreservlets.BookReader">
  <constructor-arg ref="bookLibrary"/>
</bean>
<br/>
<br/>
dean ...>
  <constructor-arg>
    st>
      <ref bean="bookLibrary"/>
      <ref bean="bookLibrary"/>
      <ref bean="bookLibrary"/>
    </list>
  </constructor-arg >
</bean>
                                          Java EE training: http://courses.coreservlets
```

General Collaborator

Flexible approach

- Supports collaborator relationships across multiple contexts
- Compatible with parent/child containers or splitting and merging contexts
- Practical for large multi-module projects

Fragile configuration

- Bean references could identify collaborator definitions supplied at runtime or errors
- No XML parser validation support
- Configuration error discovery can be delayed further if beans are lazily initialized

Avoidable risks

Risks can be mitigated through a basic and automated integration test

Java EE training: http://courses.coreservlets.com

Local Collaborator Reference

- Reference other beans in the Spring IoC container
- Leverage XML parser validation
 - Limited to collaborator beans within the same bean definitions document

14

lava EE training: http://courses.coreservlets.com

Local Collaborator Process

- Nested reference for XML constructor or property setter DI
 - XML element type ref and attribute local
 - XML property or constructor-arg child
- No short-hand equivalent
- Nested reference for XML collection element
 - For example, XML list child

15

Local Collaborator Example

16

Java EE training: http://courses.coreservlets.com

Local Collaborator

Reduces flexibility

- Presents an additional configuration constraint
 - Bean reference must identify a bean within the same document
- Mitigates the benefits of using multiple bean definition files
 - · Less practical for large multi-module projects

Improves validation

- XML parser verification support
- Bean references are verified in advance of constructing and initializing beans

Bean Identifier Reference

- Reference a literal bean identifier value
- Inject into application context support beans
 - For example, Spring AOP proxy factory beans needing to acquire the names of advisor beans
- Leverage validation support

18

Bean Identifier Process

- Nested reference for XML constructor or property setter DI
 - XML element type idref and attribute local or bean
 - Local versus bean concept consistent with local versus general bean collaborator concepts
 - local values must correlate to a bean in the same document and bean to a collaborator in the scope of the container
- Nested reference for XML collection element
 - For example, XML list child

19

Bean Identifier Example

```
<bean id="bookLibrary-00"</pre>
      class="coreservlets.JavaBookLibrary" />
<bean id="bookLibrary-01"</pre>
      class="coreservlets.JavaBookLibrary" />
<bean id="bookLibrary-02"</pre>
      class="coreservlets.JavaBookLibrary" />
<bean id="beanNames"</pre>
      class="coreservlets.BeanNames">
  <constructor-arg index="0">
    <idref local="bookLibrary-00" />
  </constructor-arg>
  <constructor-arg index="1">
    st>
      <idref bean="bookLibrary-01" />
      <idref bean="bookLibrary-02" />
    </list>
  </constructor-arg>
/bean>
                                      Java EE training: http://courses.coreservlets
```

General Approach Review

- Define and create service interfaces
- Implement services interfaces
- Add the bean definitions
 - Establish identifiers using the ia attribute
 - Aliases can be established using name attribute or alias element
 - Develop bean names consistently using a convention
 - Default to singleton beans
 - Override bean creation and caching using scope attribute
 - Specify bean inter-dependencies using DI mechanisms
 - Property setter, constructor, lookup-method, autowiring
 - Reference collaborators by referencing general, inner or local beans

General Approach Review Continued

- Access and use container-managed beans
 - The access and integration method is context-dependent

22

© 2008 coreservlets.com



Plain Value

Customized Java EE Training: http://courses.coreservlets.com/ Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Plain Value

Idea

- Insert simple values into beans without declaring separate beans for these values
- This feature actually makes Spring unique from other IoC containers

Conversion

- Values are automatically converted from String via property editors
- The conversion type is implied by the DI target
- The conversion type can be explicitly defined

24

Java EE training: http://courses.coreservlets.com

Value Declaration Process

- Create and inject plain values into multiple contexts
 - Constructor DI or property setter DI
 - Nested XML element type value
 - Short-hand value attribute
 - Nested XML element type null
 - Collection element
 - Nested XML element type value
 - Nested XML element type null

Value Declaration Example

26

Java EE training: http://courses.coreservlets.com

Value Declaration Example Continued

Null Value Declaration Example

```
<bean id="values-01" class="coreservlets.Values">
  <constructor-arg>
   <null/>
 </constructor-arg>
</bean>
<bean id="values-02" class="coreservlets.Values">
 <constructor-arg>
   st>
     <nu11/>
   </list>
 </constructor-arg>
</bean>
<bean id="values-03" class="coreservlets.Values">
 cproperty name="value">
   <nul1/>
 </bean>
```

28

Java EE training: http://courses.coreservlets.com

Value Type Conversion

- Converts values from java.lang.String
 - Uses JavaBean PropertyEditors
- Converts to implied or explicitly defined type
 - Use type attribute with a fully qualified class name for explicit declarations
- Converts to multiple types
 - Primitive and wrapper types
 - Enum types
 - List and dictionary collections
 - java.io.File
 - java.io.InputStream
 - URL format or custom Spring resource schemes
 - java.lang.Class
 - java.util.Properties

29

Value Type Conversion Example

```
public class SimpleValues {
   public enum Color{
      Red, Yellow, Blue;
   }

public void setInteger(Integer integer) {
   }

public void setColor(Color color) {
   }

public void setFile(File file) {
   }

public void setNumber(Number file) {
   }

public void setProperties(Properties properties) {
   }

public void setInputStream(InputStream inputStream) {
   }

}
```

Value Type Conversion Example

```
<beans>
  <bean class="coreservlets.SimpleValues">
    cproperty name="integer" value="100" />
    cproperty name="color" value="Red" />
    cproperty name="file" value="/etc/hosts" />
    cproperty name="inputStream"
              value="classpath:/log4j.xml" />
    property name="number">
      <value type="java.lang.Double">100</value>
    property name="properties">
      <value>
        name00=value00
        name01=value01
      </value>
    </property>
  </bean>
</beans>
                                  Java EE training: http://courses.coreservlets.cor
```

Custom Type Conversion

- Implement java.beans.PropertyEditor
 - Extend default implementationjava.beans.PropertyEditorSupport
 - Override method

setAsText(text:String):void

- Register new PropertyEditor implementation
 - See CustomEditorConfigurer from package org.springframework.beans.factory.config
 - See PropertyEditorRegistrar from package org.springframework.beans

32

Java EE training: http://courses.coreservlets.com

General Approach Review

- Define and create service interfaces
- Implement services interfaces
- Add the bean definitions
 - Establish identifiers using the ia attribute
 - Aliases can be established using name attribute or alias element
 - Develop bean names consistently using a convention
 - Default to singleton beans
 - Override bean creation and caching using scope attribute
 - Specify bean inter-dependencies using DI mechanisms
 - Property setter, constructor, lookup-method, autowiring
 - Reference collaborators by referencing inner beans or general or local beans

33

General Approach Review Continued

- Add the bean definitions continued
 - Inject plain values using the **value** attribute or element
- Access and use container-managed beans
 - The access and integration method is context-dependent

34





Properties Integration

Customized Java EE Training: http://courses.coreservlets.com/

Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Properties Integration

Injects

- Configuration properties directly into bean definitions
 - Constructor DI
 - · Property setter DI
 - · Collection element

Replaces

- Configuration management libraries
- Calls from business logic to configuration management APIs

Supports

Iterative exposure over application settings

Clarifies

Distinctions between administerable application settings and developer settings

Java EE training: http://courses.coreservlets.cor

36

Properties Integration

Feature

- Configuration mappings are supported by property editor support
- Configuration is visible from anywhere in the container

Limitation

Each container only supports one properties declaration

Properties Integration Process

- spring-context XML namespace
 - Add **spring-context** XML namespace information
- property-placeholder XML element
 - Add a property-placeholder declaration
- location attribute
 - Reference a resource path such as a classpath or filesystem location
- \${name} expression
 - Reference property file properties with the property key nested within the expression format \${name}
- Configuration management
 - For projects requiring multiple bean definition files, isolate the propertyplaceholder declaration in its own file
 - Merge the configuration file into the container with peer bean definition files

38

Java EE training: http://courses.coreservlets.com

Properties Integration Example

name.01=value

name.02=value

name.03=value

name.04=value

39

Properties Integration Example

40

Java EE training: http://courses.coreservlets.com

General Approach Review

- Define and create service interfaces
- Implement services interfaces
- Add the bean definitions
 - Establish identifiers using the ia attribute
 - Aliases can be established using name attribute or alias element
 - Develop bean names consistently using a convention
 - Default to singleton beans
 - Override bean creation and caching using scope attribute
 - Specify bean inter-dependencies using DI mechanisms
 - Property setter, constructor, lookup-method, autowiring
 - Reference collaborators by referencing inner beans or general or local beans

General Approach Review Continued

- Add the bean definitions continued
 - Inject plain values using the **value** attribute or element
 - Reference values sourced from application configuration files using the expression format \$ {name}
 - Integrate properties files using propertyplaceholder elements supplied by spring-context
- Access and use container-managed beans
 - The access and integration method is context-dependent

42

Java EE training: http://courses.coreservlets.com

© 2008 coreservlets.com



Resource Integration

Customized Java EE Training: http://courses.coreservlets.com/ Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Resource Integration

- Abstracts multiple resource types
 - Filesystem resource
 - Classpath resource
 - HTTP network resource
 - Context-specific resources
 - · For example, servlet context
- Replaces URL schemes
 - Conventional JVM URL scheme registration system is inflexible

44

Resource Integration Process

- Identify the path to the I/O source
 - For example, classpath resource path
- Rely on the default I/O context based on the ApplicationContext implementation
 - For instance, **ClassPathXmlApplicationContext** maps paths to resources located within the classpath
- Optionally, prefix the path with a custom scheme

```
- file:/path
```

- classpath:/path

- http://path

45

Resource Integration Example

46

Java EE training: http://courses.coreservlets.com

© 2008 coreservlets.com



Collections

Customized Java EE Training: http://courses.coreservlets.com/

Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Collections

- Enables list or set data structure declarations in bean definitions
- Supports data structure type conversion to target collection type
- Provides element type conversion to generic and array type

Collections

- Configuration
 - XML element type list or set
- Collection type conversion
 - java.util.Collection
 - java.util.List
 - java.util.Set
 - Array types
- Element type conversion
 - Generics
 - Array type
- XML child element types
 - bean, ref, idref, value, null, list, set, map, props

Collections Process

- Identify the DI target
 - For example, property setter or constructor
- Verify the parameter type
 - Must be an array type, java.util.Collection, or java.util.List, or java.util.Set
- Declare the XML element list or set
- Declare the collection elements as XML child elements

Java EE training: http://courses.coreservlets

Collection Example

```
public class CollectionValues {
  public void setVarargs(String... values) {
  public void setArray(String[] values) {
  public void setList(List<String> values) {
  }
  public void setSet(Set<String> values) {
  public void setCollection(Collection<String> values) {
  }
```

List Collection Example

```
<bean class="coreservlets.CollectionValues">
  property name="varargsValues">
    t>
      <value>abc</value>
    </list>
  </property>
  property name="arrayValues">
    st>
      <value>abc</value>
    </list>
  </property>
  property name="listValues">
    st>
      <value>abc</value>
    </list>
  </property>
              Java EE training: http://courses.coreservlets.com
k/bean>
```

Set Collection Example

```
<bean class="coreservlets.CollectionValues">
  cproperty name="varargsValues">
    <set>
      <value>abc</value>
    </set>
  </property>
  cproperty name="arrayValues">
      <value>abc</value>
    </set>
  </property>
  cproperty name="listValues">
      <value>abc</value>
    </set>
  </property>
</bean>
                                Java EE training: http://courses.coreservlets.com
```

Collection Type Conversion

- Inspects the target type
 - Collection generic type
 - Array type
- Converts the element type before assignment to the data structure
 - Conversion based on value PropertyEditor support

54

Collection Type Conversion Example

```
public class TypedListValues{
   public void setVarargsIntegers(Integer... values) {
   }
   public void setArrayFiles(File[] values) {
   }
   public void setListClasses(List<Class> values) {
   }
}
```

55

Collection Type Conversion Example

```
<beans>
  <bean class="coreservlets.TypedListValues">
    property name="varargsIntegers">
        <value>100</value>
      </list>
    </property>
    property name="arrayFiles">
      st>
        <value>/etc/hosts</value>
      </list>
    property name="listClasses">
      t>
        <value>java.lang.String</value>
      </list>
    </property>
  </bean>
</beans>
                                    Java EE training: http://courses.coreservlets.cor
```

Dictionary Collections

- Enables dictionary data structure declarations in bean definitions
- Supports data structure type conversion to target collection types
- Provides element type conversion to generic and array types

Map Collections

- Configuration
 - Data structure
 - XML element type map
 - Elements
 - XML element type entry and key elements
- Collection type conversion
 - java.util.Map
- Element type conversion
 - Generics
 - Array type
- XML child element types
 - bean, ref, idref, value, null, list, set, map, props
 Java EE training: http://courses.core

58

Map Collections Process

- Identify the DI target
 - For example, property setter or constructor
- Verify the parameter type
 - Must be an array type, java.util.Map
- Declare the XML element map
- Declare the map elements using XML child element type entry

Map Collections Example

```
<bean class="coreservlets.MappedValues">
  cproperty name="map">
    <map>
      <entry key="00" value="00" />
      <entry key="01">
        <value>01</value>
      </entry>
      <entry value-ref="02">
        <key><value>02</value></key>
      </entry>
      <entry>
        <key><value>03</value></key>
        <value>03</value>
      </entry>
    </map>
  </property>
</bean>
                                    Java EE training: http://courses.coreservlets.com
```

Map Collections Example

Properties Collections

- Enables simple dictionary data structure declarations in bean definitions to be
- Provides simplied interface for configuring property sets
 - No type conversion support

62

Properties Collections

- Configuration
 - Data structure
 - XML element type props
 - Elements
 - XML element type prop and key attribute
 - Property value is specified as the prop element content body
- Collection type conversion
 - java.util.Map
 - java.util.Properties
- Element type conversion
 - None
- XML child element types
 - None

Properties Collections Process

- Identify the DI target
 - For example, property setter or constructor
- Verify the parameter type
 - Must be an array type, java.util.Map or java.util.Properties
- Declare the XML element props
- Declare the map elements using XML child element type props

64

Java EE training: http://courses.coreservlets.com

Properties Collections Example

65

General Approach Review

- Define and create service interfaces
- Implement services interfaces
- Add the bean definitions
 - Establish identifiers using the ia attribute
 - Aliases can be established using name attribute or alias element
 - Develop bean names consistently using a convention
 - Default to singleton beans
 - Override bean creation and caching using scope attribute
 - Specify bean inter-dependencies using DI mechanisms
 - Property setter, constructor, lookup-method, autowiring
 - Reference collaborators by referencing inner beans or general or local beans

66

Java EE training: http://courses.coreservlets.com

General Approach Review Continued

- Add the bean definitions continued
 - Inject plain values using the **value** attribute or element
 - Reference values sourced from application configuration files using the expression format \$ {name}
 - Integrate properties files using propertyplaceholder elements supplied by spring-context
 - Inject collections by defining data structures directly within the bean definitions configuration
 - · List, set, map, or properties structures
- Access and use container-managed beans
 - The access and integration method is context-dependent



Wrap-up

Customized Java EE Training: http://courses.coreservlets.com/ Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

General collaborator references

- Flexible but fragile
- Process
 - XML element type ref with the bean attribute
 - XML attribute ref

Local collaborator references

- Improved validation but inflexible
- Process
 - XML element type ref with the local attribute
 - XML attribute ref

Summary Continued

Plain values

- XML element type **value** or attribute **value**
- XML attribute **type** to override implied conversion

Properties integration

- XML element **property-placeholder** declaration from the **spring-context** namespace
 - Requires the additional spring-context namespace declaration
- XML attribute **location** specifying the I/O source

70

Java EE training: http://courses.coreservlets.com

Summary Continued

Resource integration

- Facilitated inline when specifying path information using scheme prefixes **file**, **classpath**, **http**
- No prefix results in ApplicationContext dependent behavior

Collections

- XML parent element types list, set, map, and property

© 2008 coreservlets.com



Questions?

Customized Java EE Training: http://courses.coreservlets.com/ Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.