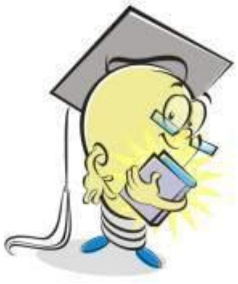
Spring’s form tag library

Cognizant

Passion for building stronger businesses



C3: Protected

About the Author

Created By:	Ramesh C.P. (161646)
Credential Information:	8+ years of experience in technical training
Version and Date:	Spring MVC/PPT/1011/3.0

Cognizant Certified Official Curriculum

UNDER THE SEAL OF COGNIZANT ACADEMY

© 2011, Cognizant Technology Solutions

Icons Used

Questions

Tools

Hands on Exercise

Coding Standards

Test Your Understanding

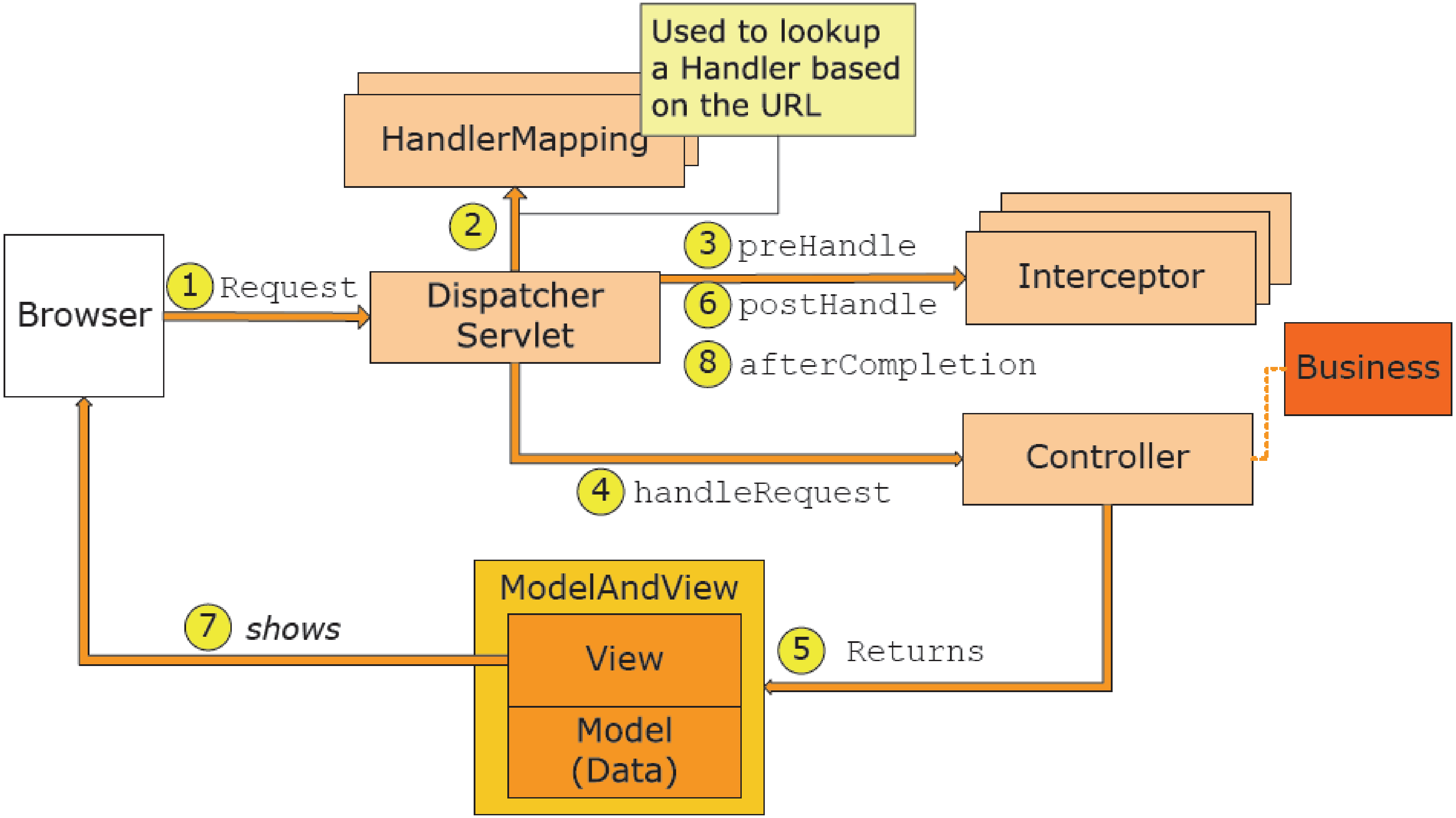
Reference

Demonstration

A Welcome Break

Contacts

Review of Architecture



© 2011, Cognizant Technology Solutions

Spring's form tag library: Overview

- Introduction:
  - Spring provides a comprehensive set of data binding-aware tags for handling form elements when using JSP and Spring Web MVC.
  - Each tag provides support for the set of attributes of its corresponding HTML tag counterpart, making the tags familiar and intuitive to use.
  - Unlike other form/input tag libraries, Spring's form tag library is integrated with Spring Web MVC, giving the tags access to the command object and reference data your controller deals with.

© 2011, Cognizant Technology Solutions

Spring’s form tag library: Objectives

- Objective:

After completing this chapter you will be able to:

  - Use spring’s form tag to develop user interface for an Web application.

Configuration

- The form tag library comes bundled in spring.jar.
- The library descriptor is called spring-form.tld.
- To use the tags from this library, add the following directive to the top of your JSP page:

  ```
  <%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
  ```

© 2011, Cognizant Technology Solutions

The Form

- The Form contains input elements to be used in the model components
  - Contains form fields bound to the command (using a Tag Library)
  - The form's `action` attribute points to the URL of the controller
  - The form's `method` must be `post` (default, can be configured differently)

```
<form action="Registration.form" method="post">
  ...
</form>
```

© 2011, Cognizant Technology Solutions

The Form (Contd.)

Tag	Description
`htmlEscape`	Global value specifying if Spring tags should be escaped (can be overridden on other tags).
`message`	Shows a message from a messageSource (similar to JSTL's `message` tag).
`theme`	Obtains a value from a theme (for Themed pages). The value is based on the current theme.
`transform`	Transforms the supplied value using a registered property editor. Used with forms.
`bind`	Binds the body to an object and property ("path") used to create status, errors and property editors. Used with forms.
`hasBindErrors`	Conditional tag whose body is evaluated when the specified bean contains errors. Used with forms.
`nestedPath`	Nested bind tags will prepend their path (similar to the `with` keyword in Pascal). Used with forms.

© 2011, Cognizant Technology Solutions

The form tag

- This tag renders an HTML 'form' tag and exposes a binding path to inner tags for binding.
- It puts the command object in the PageContext so that the command object can be accessed by inner tags.
- All the other tags in this library are nested tags of the form tag.

© 2011, Cognizant Technology Solutions

The form tag

```
<form:form commandName="user">
   <table>
     <tr>
       <td>First Name:</td>
       <td><form:input path="firstName" /></td>
     </tr>
     <tr>
       <td>Last Name:</td>
       <td><form:input path="lastName" /></td>
     </tr>
     <tr>
       <td colspan="2">
         <input type="submit" value="Save Changes" />
       </td>
     </tr>
   </table>
 </form:form>
```

The Form (Contd.)

```
<%@ taglib uri="http://www.springframework.org/tags" prefix="spring" %>
<form action="Registration.form" method="post">
  <table>
    <tr>
      <td>Username</td>
      <td>
        <spring:bind path="command.username">
          <input name="username" value="${status.value}"/>
          ${status.errorMessage}
        </spring:bind>
      </td>
   </tr>
   <tr>
     <td>email</td>
     <td>
       <spring:bind path="command.email">
         <input name="email" value="${status.value}"/>
         ${status.errorMessage}
       </spring:bind>
     </td>
    </tr>
    ...
  </table>
</form>
```

© 2011, Cognizant Technology Solutions

The Spring Form tags

- SpringMVC has a number of custom tags that can help assist in page rendering
  - Prior to Spring 2.0, the <spring:bind> tag was used to bind a command object to a form field
  - This technique was cumbersome and was improved
- Use <form:tagname> tag library with the following taglib declaration:

```
<%taglib prefix="form"
    uri="http://www.springframework.org/tags/form"%>
```

© 2011, Cognizant Technology Solutions

The Spring Form tags

- The Spring Custom Form Tags:

  form
  input
  label
  errors
  textarea
  checkbox
  checkboxes

  select
  password
  hidden
  radiobutton
  radiobuttons
  option
  options

© 2011, Cognizant Technology Solutions

Using a `PropertyEditor`

- Certain data types must be rendered and entered in a specific way
  - Spring uses the standard `java.beans.PropertyEditor` for this
  - Registered inside the controller for a specific type (inside the `initBinder` method)

```
protected void initBinder(HttpServletRequest request,
        ServletRequestDataBinder binder) throws Exception {
  SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd",
  request.getLocale());
  binder.registerCustomEditor(Date.class,new
                              CustomDateEditor(df,false));
}
```

The input tag

- This tag renders an HTML 'input' tag with type 'text' using the bound value

```
<form:input path="firstName" />
<form:input path=“lastName”  />
```

The checkbox tag

- This tag renders an HTML 'input' tag with type 'checkbox'.

```
 // User preference class
  public class Preferences {
        private String[] interests;
        // getter and setter methods
       ...
 }
```

The checkbox tag (Contd.)

```
<form:form commandName=“user”>
<table>
…
 <tr>
        <td>Interests:</td>
        <td>
          Java 5:       <form:checkbox path="preferences.interests" value=“Java 5"/>
          Pro Spring: <form:checkbox path="preferences.interests" value=“Pro Spring"/>
           JSF Pro:      <form:checkbox path="preferences.interests” value=“JSF Pro"/>
        </td>
</tr>
…
</table>
</form:form>
```

© 2011, Cognizant Technology Solutions

The radiobutton tag

- This tag renders an HTML 'input' tag with type 'radio'.

```
<tr>
    <td>Sex:</td>
    <td>Male: <form:radiobutton path="sex" value="M"/>
   <br/>
    Female: <form:radiobutton path="sex" value="F"/>
   </td>
   <td>&nbsp;</td>
 </tr>
```

The password tag

- This tag renders an HTML 'input' tag with type 'password' using the bound value.

```
 <tr>
     <td>Password:</td>
     <td> <form:password path="password" /> </td>
</tr>
```

The select tag

- This tag renders an HTML 'select' element:

Let us assume a User has a list of skills

```
<tr>
        <td>Skills:</td>
        <td><form:select path="skills" items="${skills}"/></td>
 </tr>
```

The option tag

- This tag renders an HTML 'option‘

```
  <tr>
   <td>City:</td>
   <td>
        <form:select path="city">
               <form:option value="Chennai"/>
               <form:option value="Los Angles"/>
               <form:option value="Mumbai"/>
               <form:option value="Pune"/>
   s       </form:select>
  </td>
</tr>
```

© 2011, Cognizant Technology Solutions

The textarea tag

- This tag renders an HTML 'textarea‘

```
 <tr>
   <td>Comments :</td>
   <td><form:textarea path=“comments" rows="3" cols="20“ /></td>
</tr>
```

The Hidden tag

- This tag renders an HTML 'input' tag with type 'hidden' using the bound value.
- To submit an unbound hidden value, use the HTML input tag with type 'hidden'.

```
<form:hidden path=“todo" />
```

The errors tag

- This tag renders field errors in an HTML 'span' tag.
- It provides access to the errors created in your controller or those that were created by any validators associated with your controller.

```
<form:input path="userName" />
   <%-- Show errors for userName field --%>
 <form:errors path="userName" />
```

path="*" - displays all errors

Path=“userName” - displays all errors associated with the userName field

Spring's form tag library

Time for a Break !

Spring's form tag library

- Questions from participants

© 2011, Cognizant Technology Solutions

Test Your Understanding

1) Which of the following is a mandatory attribute of form input tag

   a) path
   b) id
   c) lang
   d) Title

2) The tag-generated HTML is HTML 4.01/XHTML 1.0 compliant. Say true or false

© 2011, Cognizant Technology Solutions

Spring's form tag library: Summary

- Spring 3.0 provides the very powerful tag library to create user interface using JSPs
- The tag library consists of tags to generate HTML tags that are HTML 4.X compliant.
- These tags are data-binding aware.

Spring's form tag library : Source

- http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html/view.html#view-jsp-formtaglib

Disclaimer: Parts of the content of this course is based on the materials available from the Web sites and books listed above. The materials that can be accessed from linked sites are not maintained by Cognizant Academy and we are not responsible for the contents thereof. All trademarks, service marks, and trade names in this course are the marks of the respective owner(s).

You have successfully completed Spring’s form tag library

Click here to proceed

Cognizant

Passion for building stronger businesses