



Apache Struts: An MVC Framework

Lecture 2: Prepopulating and Redisplaying HTML Forms

Core Servlets & JSP book: www.coreservlets.com
More Servlets & JSP book: www.moreservlets.com
Servlet and JSP Training Courses: courses.coreservlets.com

Slides © Marty Hall, <http://www.coreservlets.com>, books © Sun Microsystems Press

Overview

- Prepopulating HTML forms
- Using the `html:form` tag
- Using `html:text` and other related tags
- Redisplaying forms
- Applying session tracking
- Adding warning messages

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Prepopulating HTML Forms

- **Based on data in a Java object, how do you:**
 - Choose names?
 - Remember that parameter names must match bean properties
 - Fill in default values?
- **Motivation**
 - Objects have the real data
 - Java object could, in turn, come from a database
 - You want values to automatically change as code changes
- **Steps**
 - Set up a form with `<html:form ...>`
 - Automatically associated with a bean listed in `struts-config.xml`
 - Set up input elements with `<html:xxx property="blah"/>`
 - NAME taken directly from property name
 - VALUE is the `blah` property of the bean

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Defining a Form

1. **Use the struts-html tag library**
`<% @ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>`
2. **Use `<html:form ...>` instead of `<FORM ...>`**
 - Choose `action` that is relative to the Web application
 - `.do` implied automatically
 - The address of the JSP page that has `<html:form>` must be listed as the `input` attribute of an `action` entry in `struts-config.xml`
 - Use other tag attributes the same as with `FORM`
 - Except that default method is `POST`, not `GET`
 - Attributes accept request-time expressions (`<%= ... %>`)
 - **Result:** bean automatically associated with the form and created when the JSP page is accessed.

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Defining a Form: Details

- **struts-config.xml:**

```
<form-bean name="beanName" type="..."/> ...
<action path="/actions/blah"
  type="..."
  name="beanName" scope="..."
  input="/somePath/somePage.jsp">...</action>
```
- **webAppDir/somePath/somePage.jsp:**

```
<html:form action="/actions/blah">
- Note that the action is not /webAppPrefix/actions/blah
- Note also that the action is not .../blah.do
- Resultant HTML:
<FORM ACTION="/webAppPrefix/actions/blah.do"
  METHOD="POST">
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Creating Input Elements

1. **Use `<html:text property="blah" />` for textfields**
 - NAME becomes "blah"
 - VALUE becomes `theBean.getBlah()`
 - I.e., the following are equivalent (after you create bean)
 - `<html:text property="email"/>`
 - `<INPUT TYPE="TEXT" NAME="email" VALUE="<%= theBean.getEmail() %>">`
2. **Use `<html:xxx.../>` for other input elements**
 - `<html:submit.../>`
 - `<html:password .../>`
 - `<html:radio .../>`
 - `<html:checkbox .../>`
 - Normal attributes supported, and you can override value.

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 1: Editing struts-config.xml

```
...
<form-beans>
  <form-bean name="contactFormBean"
    type="coreservlets.ContactFormBean"/>
</form-beans>

<action-mappings>
  <action path="/actions/signup1"
    type="coreservlets.Signup1"
    name="contactFormBean"
    scope="request"
    input="/forms/signup1.jsp">
    <forward name="missing-value"
      path="/WEB-INF/results/missing-value.jsp"/>
    <forward name="success"
      path="/WEB-INF/results/confirmation.jsp"/>
  </action>
</action-mappings>
...
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 2: Create Form Bean

D:\Java\Struts1.1\struts-test4\WEB-INF\src\java\coreservlets

```
package coreservlets;
import org.apache.struts.action.*;

public class ContactFormBean extends ActionForm {
  private String firstName = "First name";
  private String lastName = "Last name";
  private String email = "user@host";
  private String faxNumber = "xxx-yyy-zzzz";

  public String getFirstName() {
    return(firstName);
  }

  public void setFirstName(String firstName) {
    this.firstName = firstName;
  }
  ...
}
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 3: Create Results Beans

D:\Java\Struts1.1\struts-test4\WEB-INF\src\java\coreservlets

```
package coreservlets;

public class MessageBean {
  private String message = "";

  public String getMessage() {
    return(message);
  }

  public void setMessage(String message) {
    this.message = message;
  }
}
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 4: Create Action Subclass

D:\Java\Struts1.1\struts-test4\WEB-INF\src\java\coreservlets

```
package coreservlets;

import javax.servlet.http.*;
import org.apache.struts.action.*;

public class Signup1 extends Action {
  public ActionForward execute(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response)
    throws Exception {
    ContactFormBean userBean = (ContactFormBean)form;
    String firstName = userBean.getFirstName();
    String lastName = userBean.getLastName();
    String email = userBean.getEmail();
    String faxNumber = userBean.getFaxNumber();
    MessageBean message = new MessageBean();
    request.setAttribute("messageBean", message);
  }
}
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 4: Create Action Subclass (Continued)

```
if (isMissing(firstName)) {
  message.setMessage("first name");
  return(mapping.findForward("missing-value"));
} else if (isMissing(lastName)) {
  message.setMessage("last name");
  return(mapping.findForward("missing-value"));
} else if ((isMissing(email) ||
  (email.indexOf("@") == -1)) {
  message.setMessage("email address");
  return(mapping.findForward("missing-value"));
} else if (isMissing(faxNumber)) {
  message.setMessage("fax number");
  return(mapping.findForward("missing-value"));
} else {
  return(mapping.findForward("success"));
}

private boolean isMissing(String value) {
  return((value == null) || (value.trim().equals("")));
}
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 5: Create Form

D:\Java\Struts1.1\struts-test4\forms\signup1.jsp

```
<!DOCTYPE ...>
...
<CENTER>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<html:form action="/actions/signup1">
  First name: <html:text property="firstName"/><BR>
  Last name: <html:text property="lastName"/><BR>
  Email address: <html:text property="email"/><BR>
  Fax number: <html:text property="faxNumber"/><BR>
  <html:submit value="Sign Me Up!"/>
</html:form>
</CENTER>
</BODY></HTML>
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 5: Create Form

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 6: Display Results in JSP

D:\Java\Struts1.1\struts-test4\WEB-INF\results\missing-value.jsp

```
<!DOCTYPE ...>
<HTML>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<HEAD><TITLE>Missing
<bean:write name="messageBean" property="message"/>
</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<CENTER>
<H2>Missing
<bean:write name="messageBean" property="message"/>!</H2>
Please <A HREF="../forms/signup1.jsp">try again</A>.
</CENTER>
</BODY></HTML>
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 6: Display Results in JSP

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 6: Display Results in JSP

D:\Java\Struts1.1\struts-test4\WEB-INF\results\confirmation.jsp

```
<!DOCTYPE ...>
<HTML>
<HEAD><TITLE>Confirmation</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<CENTER>
<H1>Confirmation</H1>
Congratulations. You are now signed up for the
Single Provider of Alert Memos network!
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<UL>
<LI>First name:
<bean:write name="contactFormBean" property="firstName"/>
<LI>Last name:
<bean:write name="contactFormBean" property="lastName"/>
<LI>Email address:
<bean:write name="contactFormBean" property="email"/>
<LI>Fax number:
<bean:write name="contactFormBean" property="faxNumber"/>
</UL>
To be removed from the network, send email
<A HREF="mailto:blackhole@spam-network.com">here</A>.
</CENTER>
</BODY></HTML>
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 6: Display Results in JSP

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Redisplaying HTML Forms

- When user submits incomplete form, how do you:
 - Redisplay form?
 - Action shouldn't need to know address of the form
 - Fill in previously entered values?
- Steps
 1. Declare form bean to be session-scoped
 2. Return an "incomplete" message when form bean is missing some required fields
 3. Store warning message in form bean
 4. Map "incomplete" back to the original form
 5. Display warning message in original form with bean:write
 - Use filter="false" if warning message contains HTML tags

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 1: Editing struts-config.xml

```
...
<form-beans>
  <form-bean name="contactFormBean"
            type="coreservlets.ContactFormBean"/>
</form-beans>
<action-mappings>
  <action path="/actions/signup2"
        type="coreservlets.Signup2"
        name="contactFormBean"
        scope="session"
        input="/forms/signup2.jsp">
    <forward name="missing-value"
            path="/forms/signup2.jsp"/>
    <forward name="success"
            path="/WEB-INF/results/confirmation.jsp"/>
  </action>
</action-mappings>
...
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 2: Create Form Bean (Update to Previous Code)

```
public class ContactFormBean extends ActionForm {
  ...
  private String warning = "";

  public String getWarning() {
    return(warning);
  }

  public void setWarning(String baseWarning) {
    this.warning =
      "<H2><FONT COLOR=RED>Missing " +
      baseWarning + "!</FONT></H2>";
  }
}
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 3: Create Results Beans

- Not used in this example

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 4: Create Action Subclass

```
public class Signup2 extends Action {
  public ActionForward execute(ActionMapping mapping,
                              ActionForm form,
                              HttpServletRequest request,
                              HttpServletResponse response)
    throws Exception {
    ContactFormBean contactBean = (ContactFormBean)form;
    String firstName = contactBean.getFirstName();
    String lastName = contactBean.getLastName();
    String email = contactBean.getEmail();
    String faxNumber = contactBean.getFaxNumber();
  }
}
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 4: Create Action Subclass (Continued)

```
if (isMissing(firstName)) {
  contactBean.setWarning("first name");
  return(mapping.findForward("missing-value"));
} else if (isMissing(lastName)) {
  contactBean.setWarning("last name");
  return(mapping.findForward("missing-value"));
} else if ((isMissing(email) ||
            (email.indexOf("@") == -1)) {
  contactBean.setWarning("email address");
  return(mapping.findForward("missing-value"));
} else if (isMissing(faxNumber)) {
  contactBean.setWarning("fax number");
  return(mapping.findForward("missing-value"));
} else {
  return(mapping.findForward("success"));
}

private boolean isMissing(String value) {
  return((value == null) || (value.trim().equals("")));
}
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 5: Create Form

```
<!DOCTYPE ...>
...
<CENTER>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<html:form action="/actions/signup2">
  <bean:write name="contactFormBean" property="warning"
            filter="false" />
  First name: <html:text property="firstName"/><BR>
  Last name: <html:text property="lastName"/><BR>
  Email address: <html:text property="email"/><BR>
  Fax number: <html:text property="faxNumber"/><BR>
  <html:submit value="Sign Me Up!"/>
</html:form>
</CENTER>
</BODY></HTML>
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 5: Create Form

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 6: Display Results in JSP (Same Source Code as Form)

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 6: Display Results in JSP (Same Code as Previous Example)

```
<!DOCTYPE ...>
<HTML>
<HEAD><TITLE>Confirmation</TITLE></HEAD>
<BODY BGCOLOR="#FDF5B6">
<CENTER>
<H1>Confirmation</H1>
Congratulations. You are now signed up for the
Single Provider of Alert Memos network!
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<UL>
<LI>First name:
<bean:write name="contactFormBean" property="firstName"/>
<LI>Last name:
<bean:write name="contactFormBean" property="lastName"/>
<LI>Email address:
<bean:write name="contactFormBean" property="email"/>
<LI>Fax number:
<bean:write name="contactFormBean" property="faxNumber"/>
</UL>
To be removed from the network, send email
<A HREF="mailto:blackhole@spam-network.com">here</A>.
</CENTER>
</BODY></HTML>
```

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Step 6: Display Results in JSP

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com

Summary

- Use `<html:form ...>` instead of `<FORM ...>`
 - action URLs interpreted relative to Web application
 - Bean automatically associated with the form
 - Bean class taken from whatever action this page is listed as input of
- Use `<html:text property="blah" />` for textfields
 - NAME becomes "blah"
 - VALUE becomes theBean.getBlah()
- For form redisplay, declare bean to be session-scoped
- Output custom warnings with bean:write
 - Remember to use filter="false" if warning contains HTML tags

Apache Struts Lecture 2: Handling Forms

www.coreservlets.com



Questions?

Core Servlets & JSP book: www.coreservlets.com
 More Servlets & JSP book: www.moreservlets.com
 Servlet and JSP Training Courses: courses.coreservlets.com

Slides © Marty Hall, <http://www.coreservlets.com>, books © Sun Microsystems Press