

Apache Struts: An MVC Framework

Lecture 1: Basics

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


Servlet-based Frameworks

- **Struts**
 - A Sun-proposed defacto standard for J2EE
 - Widely popular among server-side Java developers
 - If JSPs are heavily used, Struts may be the best choice for enforcing an MVC architecture
- **Velocity**
 - Apache project that provides a passive presentation layer with PHP-like syntax and semantics
 - Unlike JSPs (which are active components), Velocity *templates* (also called *macros*) are passive components (so MVC is preserved)
- **Many others abound**
 - http://www.frameworks-boulder.org/Application_Frameworks.html

Apache Struts Lecture 1: Intro www.coreservlets.com

JavaServer Faces (JSF)

- Sun's effort to define a standard, visual server-side framework for servlets, JSPs, JSTL, etc.
 - Highly decoupled, plug-in support for other frameworks
 - JSF will be part of the J2EE 1.5 standard (early 2005?)



Apache Struts Lecture 1: Intro www.coreservlets.com

Overview

- **Understanding Struts**
- **Getting ready**
 - Installing Struts
 - Configuring your environment
 - Testing Struts
- **Setting up a Struts application**
- **Using Struts**
 - Using `struts-config.xml` to map `blah.do` to Actions
 - Creating Actions
 - Creating a form that invokes `blah.do`
 - Defining form beans to be populated by form submission
 - Creating data/results beans
 - Displaying results in JSP

Apache Struts Lecture 1: Intro www.coreservlets.com

Struts: Advantages

- **Configuration via XML files**
 - Many changes can be made without modifying or recompiling Java code
 - Pages are loosely coupled; URLs not hardcoded
- **Form beans**
 - Greatly simplifies processing of request parameters
- **HTML tags**
 - Lets you get initial values from Java objects
 - Lets you redisplay forms with previous values intact
- **Form field validation**
 - Client-side and server-side
- **Consistent implementation**
 - Encourages consistent use of MVC throughout applications which employ JSP

Apache Struts Lecture 1: Intro www.coreservlets.com

Struts: Advantages

- **Internationalization and Localization**
 - I18N and L10N
 - Eg. en-us, en-ca, fr-ca
 - Supported through “resources”
 - Usually just a “properties” file
 - i.e. `resources.properties`

Struts

Apache Struts Lecture 1: Intro www.coreservlets.com

Struts: Disadvantages

- **Bigger learning curve**
 - Need to know JSP/servlet APIs plus a large framework
- **Worse documentation**
 - Fewer books and less-clear specs than core servlet/JSP library (but that's changing)
- **Less transparent**
 - More behind-the-scenes tasks going on
 - Harder to understand
 - Harder to benchmark and optimize
- **Forces you to think "their way"**
 - Flip side is "encourages consistent approaches"

Apache Struts Lecture 1: Intro

www.coreservlets.com

Downloading and Configuring Struts

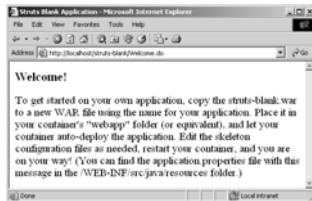
- **Download the Struts zip file**
 - Start at <http://jakarta.apache.org/site/binindex.cgi> or follow link from <http://jakarta.apache.org/struts/>
 - I'm using Struts 1.1
- **Unzip into a directory of your choice**
 - E.g., C:\jakarta-struts-1.1
 - Hereafter referred to as *struts_install_dir*
- **Update your CLASSPATH**
 - Add *struts_install_dir/lib/struts.jar*
- **JDK 1.2/1.3 (non-Tomcat): Install an XML parser**
 - One already comes with Tomcat and with JDK 1.4
 - Two popular free parsers:
 - <http://java.sun.com/xml/>
 - <http://xml.apache.org/>

Apache Struts Lecture 1: Intro

www.coreservlets.com

Testing Struts

- **Install struts-blank.war**
 - Copy *struts_install_dir/webapps/struts-blank.war* to *tomcat_install_dir/webapps/*.
- **Start the server**
- **Access <http://localhost/struts-blank/>**



Apache Struts Lecture 1: Intro

www.coreservlets.com

Setting Up a Struts Application

- **Copy/rename struts-blank**
 - Copy *tomcat_install_dir/webapps/struts-blank/* to the location you use for developing custom Web apps
 - Rename it
 - I'll call it *devel_dir/struts-test/*
- **Alternative approach (equally good)**
 - Copy *struts_install_dir/webapps/struts-blank.war* to your development directory
 - Make a new directory called *struts-test*
 - Unjar *struts-blank.war* into that directory
 - You can use "jar -xvf" or a zip tool like WinZip
- **Common approach (bad)**
 - Work directly in server's deployment directory

Apache Struts Lecture 1: Intro

www.coreservlets.com

Struts Documentation

- **Install documentation Web app**
 - *struts_install_dir/webapps/struts-documentation.war*
 - On Tomcat, just copy file to *tomcat_install_dir/webapps/*, restart server, and use the URL <http://host/struts-documentation/>
- **Bookmark <http://jakarta.apache.org/struts/learning.html>**
 - The most up-to-date info
 - FAQs
 - User guide
 - Tutorials
 - Javadoc

Apache Struts Lecture 1: Intro

www.coreservlets.com

Using Struts

1. **Use WEB-INF/struts-config.xml to:**
 - Designate Action classes to handle requests for *blah.do*
 - Specify URLs that apply in various situations
 - Declare any form beans that are being used
2. **Create form bean to be populated by the form submission**
3. **Create other results beans**
4. **Create Action subclasses to handle requests**
5. **Create form that invokes *blah.do***
 - `<FORM ACTION=".../blah.do" ...>...</FORM>`
6. **Display results in JSP**
 - Usually uses Struts bean or HTML tags
 - Sometimes uses JSTL or Struts looping/logic tags

Apache Struts Lecture 1: Intro

www.coreservlets.com

Examples

- **Example 1**
 - One simple result mapping
 - No beans
- **Example 2**
 - Multiple result mappings
 - No beans
- **Example 3**
 - Multiple result mappings
 - Form and results beans
 - Struts tags

Apache Struts Lecture 1: Intro

www.coreservlets.com

Example 1

- **Components**
 - One simple result mapping
 - No beans
- **Illustrates**
 - Editing struts-config.xml
 - Creating an Action
 - Forms that invoke Actions
 - Application organization

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 1: Editing struts-config.xml

1. **Make an action entry within action-mappings**
 - **path**: URL pattern that should invoke the Action
 - ".do" implied! You say `.../blah` but real pattern is `.../blah.do`
 - **type**: fully qualified class name of Action
 - **scope**: as in normal MVC (request, session, application)
 - **name**: bean name (use arbitrary name if no beans)
 - **input**: form that will trigger the Action
2. **Create one or more forward entries within action**
 - **name**: String that will be returned from the Action
 - **path**: address of associated JSP page
 - **redirect**: forward if false (default); redirect if true
3. **Restart server**
 - struts-config.xml read only when application starts

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 1: Editing struts-config.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE ...>
<struts-config>
  <action-mappings>
    <action path="/actions/register1" (really register1.do)
            type="coreservlets.RegisterAction1"
            name="noBean"
            scope="request"
            input="/forms/register1.jsp">
      <forward name="success"
              path="/WEB-INF/results/result1.jsp"/>
    </action>
    ...
  </action-mappings>
</struts-config>
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Steps 2 and 3

- **Bean to be populated by form submission**
 - Values automatically filled in based on correspondence between request parameter names and bean property names
 - Like `<jsp:setProperty property="*" .../>` or my `BeanUtils.populateBean` method
- **Results beans**
 - Objects to hold results of business logic, data access logic, etc.
 - Just like the beans in normal MVC
- **Beans will be discussed/illustrated in later examples**

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 4: Create Action Subclass

1. **Add Struts-specific import statements**

```
import javax.servlet.http.*;
import org.apache.struts.action.*;
```
2. **Extend the Action class**

```
public class SomeAction extends Action { ... }
```
3. **Override the execute method**

```
public ActionForward execute(ActionMapping mapping,
                             ActionForm form,
                             HttpServletRequest request,
                             HttpServletResponse response)
    throws Exception {
```
4. **Return mapping.findForward**

```
return(mapping.findForward("name-matching-entry-in-forward"));
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 4: Create Action Subclass

```
D:\Java\Struts1.1\struts-test\WEB-INF\src\java\coreservlets
package coreservlets;

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

public class RegisterAction1 extends Action {
    public ActionForward execute(ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        return(mapping.findForward("success"));
    }
}
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 5: Create Form

- **ACTION should list ../blah.do**
 - You must explicitly include the .do part, even though you omit it in the struts-config.xml file
- **Relative URLs should be used**
 - So you can develop on one machine and deploy on another without editing the addresses
 - Remember that browser, not server, interprets relative URLs in ACTION
 - If form is not in top-level dir, use ".." or "/appName" in URL
- **Struts-specific custom tags often used**
 - Lets you associate a bean with the input elements
 - Make initial values come from database
 - Redisplay partially completed form
 - Discussed/illustrated in later lectures

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 5: Create Form

```
D:\Java\Struts1.1\struts-test\forms\register1.jsp
<!DOCTYPE ...>
<HTML>
<HEAD><TITLE>New Account Registration</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<CENTER>
<H1>New Account Registration</H1>
<FORM ACTION="../actions/register1.do" METHOD="POST">
    Email address: <INPUT TYPE="TEXT" NAME="email"><BR>
    Password: <INPUT TYPE="PASSWORD" NAME="password"><BR>
    <INPUT TYPE="SUBMIT" VALUE="Sign Me Up!">
</FORM>
</CENTER>
</BODY></HTML>
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 5: Create Form



Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 6: Display Results in JSP

- **System automatically forwards to appropriate page**
 - By matching condition returned by the action to the name listed in the forward entry in struts-config.xml
- **Displaying results beans in JSP**
 - Discussed/illustrated in later examples

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 6: Display Results in JSP

```
D:\Java\Struts1.1\struts-test\WEB-INF\results\result1.jsp
<!DOCTYPE ...>
<HTML>
<HEAD><TITLE>Success</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<CENTER>
<H1>You have registered successfully.</H1>
(Version 1)
</CENTER>
</BODY></HTML>
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 6: Display Results in JSP



Apache Struts Lecture 1: Intro

www.coreservlets.com

Example 1: Organization

- **Option 1**
 - Action path gives a name only
 - `<action path="/register1" ...>`
 - Form placed in top-level Web application directory
 - `http://hostname/struts-test/register1.jsp`
 - Form specifies only *blah.do*
 - `<FORM ACTION="register1.do" ...>`
- **Option 2**
 - Action path gives a pseudo-directory name
 - `<action path="/actions/register1" ...>`
 - Form placed in Web application subdirectory
 - `http://hostname/struts-test/forms/register1.jsp`
 - Form specifies path with `"/"` or `"/webAppName/..."`
 - `<FORM ACTION="./actions/register1.do" ...>`
- **Why is second option preferable?**

Apache Struts Lecture 1: Intro

www.coreservlets.com

Example 2

- **Components**
 - Multiple result mappings
 - No beans
- **Illustrates**
 - Editing struts-config.xml
 - Multiple forward entries within the action element
 - Creating an Action
 - Multiple mapping.findForward calls

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 1: Editing struts-config.xml

```
...
<action-mappings>
  <action path="/actions/register2"
          type="coreservlets.RegisterAction2"
          name="noBean"
          scope="request"
          input="/forms/register2.jsp">
    <forward name="success"
            path="/WEB-INF/results/result2.jsp"/>
    <forward name="bad-address"
            path="/WEB-INF/results/bad-address2.jsp"/>
    <forward name="bad-password"
            path="/WEB-INF/results/bad-password2.jsp"/>
  </action>
  ...
</action-mappings>
...
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Steps 2 and 3: Creating Beans

- Postponed until next example

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 4: Create Action Subclass

```
D:\Java\Struts1.1\struts-test2\WEB-INF\src\java\coreservlets
public class RegisterAction2 extends Action {
  public ActionForward execute(ActionMapping mapping,
                              ActionForm form,
                              HttpServletRequest request,
                              HttpServletResponse response)
    throws Exception {
    String email = request.getParameter("email");
    String password = request.getParameter("password");
    if ((email == null) ||
        (email.trim().length() < 3) ||
        (email.indexOf("@") == -1)) {
      return(mapping.findForward("bad-address"));
    } else if ((password == null) ||
               (password.trim().length() < 6)) {
      return(mapping.findForward("bad-password"));
    } else {
      return(mapping.findForward("success"));
    }
  }
}
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 5: Create Form

D:\Java\Struts1.1\struts-test2\forms\register2.jsp

```
<!DOCTYPE ...>
<HTML>
<HEAD><TITLE>New Account Registration</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<CENTER>
<H1>New Account Registration</H1>
<FORM ACTION=".." />
  Email address: <INPUT TYPE="TEXT" NAME="email"><BR>
  Password: <INPUT TYPE="PASSWORD" NAME="password"><BR>
  <INPUT TYPE="SUBMIT" VALUE="Sign Me Up!">
</FORM>
</CENTER>
</BODY></HTML>
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

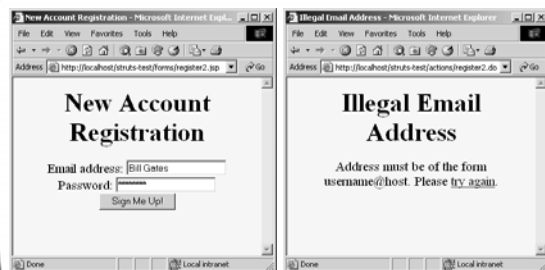
Step 5: Create Form



Apache Struts Lecture 1: Intro

www.coreservlets.com

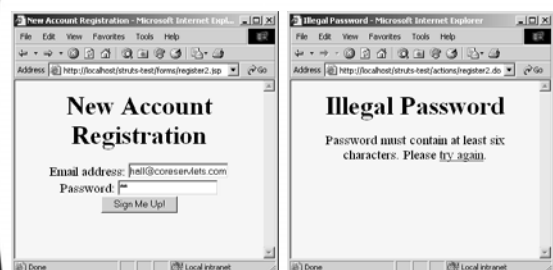
Step 6: Display Results in JSP



Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 6: Display Results in JSP



Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 6: Display Results in JSP



Apache Struts Lecture 1: Intro

www.coreservlets.com

Example 3

- **Components**
 - Multiple result mappings
 - Form bean to hold request data
 - Result bean to hold computed data
- **Illustrates**
 - Editing struts-config.xml
 - form-bean declaration
 - action name matching form-bean name
 - Creating an Action
 - Typecast of incoming form bean
 - Creating and storing results beans

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 1: Editing struts-config.xml

```
...
<form-beans>
  <form-bean name="userFormBean"
    type="coreservlets.UserFormBean"/>
</form-beans>
<action-mappings>
  <action path="/actions/register3"
    type="coreservlets.RegisterAction3"
    name="userFormBean"
    scope="request"
    input="/forms/register3.jsp">
    <forward name="success"
      path="/WEB-INF/results/result3.jsp"/>
    <forward name="bad-address"
      path="/WEB-INF/results/bad-address3.jsp"/>
    <forward name="bad-password"
      path="/WEB-INF/results/bad-password3.jsp"/>
  </action>
...
</action-mappings>
...

```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 2: Create Form Bean to be Autopopulated

- **Must extend ActionForm**
 - Argument to "execute" will be of type ActionForm
 - You should cast value to your real type
 - ActionForm is in org.apache.struts.action package
 - Alternatively, you can use DynaActionForm, which results in a Map of incoming names and values
- **Must have a zero argument constructor**
 - System will automatically call this default constructor
- **Must have setBlah methods**
 - One method corresponding to each incoming request parameter that you want inserted automatically
 - Request param names must match bean property names
- **Must have getBlah methods**
 - One method corresponding to each bean property that you want to display in JSP without Java syntax

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 2: Create Form Bean

D:\Java\Struts1.1\struts-test3\WEB-INF\src\java\coreservlets

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

public class UserFormBean extends ActionForm {
  private String email = "Missing address";
  private String password = "Missing password";

  public String getEmail() {
    return(email);
  }
  public void setEmail(String email) {
    this.email = email;
  }

  public String getPassword() {
    return(password);
  }
  public void setPassword(String password) {
    this.password = password;
  }
}

```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 3: Create Results Beans

- **These are the normal value objects used in MVC**
 - Do *not* need to extend any particular class
 - Do *not* need zero argument constructors if JSP page will never create the objects
 - Still need getter methods
 - Setter methods are optional; you sometimes supply all needed values to the constructor
 - Often returned by business-logic or data-access-logic code

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 3: Create Results Beans

D:\Java\Struts1.1\struts-test3\WEB-INF\src\java\coreservlets

```
package coreservlets;

public class SuggestionBean {
  private String email;
  private String password;

  public SuggestionBean(String email, String password) {
    this.email = email;
    this.password = password;
  }

  public String getEmail() {
    return(email);
  }

  public String getPassword() {
    return(password);
  }
}

```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 4: Create Action Subclass

D:\Java\Struts1.1\struts-test3\WEB-INF\src\java\coreservlets

```
public class RegisterAction3 extends Action {
  public ActionForward execute(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response)
    throws Exception {
    UserFormBean userBean = (UserFormBean)form;
    String email = userBean.getEmail();
    String password = userBean.getPassword();
    SuggestionBean suggestionBean =
      SuggestionUtils.getSuggestionBean();
    request.setAttribute("suggestionBean", suggestionBean);
    if ((email == null) ||
      (email.trim().length() < 3) ||
      (email.indexOf("@") == -1)) {
      return(mapping.findForward("bad-address"));
    } else if ((password == null) ||
      (password.trim().length() < 6)) {
      return(mapping.findForward("bad-password"));
    } else {
      return(mapping.findForward("success"));
    }
  }
}

```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 5: Create Form

D:\Java\Struts1.1\struts-test3\forms\register3.jsp

```
<!DOCTYPE ...>
<HTML>
<HEAD><TITLE>New Account Registration</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<CENTER>
<H1>New Account Registration</H1>
<FORM ACTION=".."actions/register3.do" METHOD="POST">
  Email address: <INPUT TYPE="TEXT" NAME="email"><BR>
  Password: <INPUT TYPE="PASSWORD" NAME="password"><BR>
  <INPUT TYPE="SUBMIT" VALUE="Sign Me Up!">
</FORM>
</CENTER>
</BODY></HTML>
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 5: Create Form



Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 6: Display Results in JSP

- **Use JSP scripting elements**
 - Untenable! This is what Struts is designed to avoid.
- **Use `jsp:useBean` and `jsp:getProperty`**
 - Possible, but a little verbose and clumsy
- **Use JSTL `c:out` tag**
 - Not worth the bother unless you are already using JSTL
- **Use JSP 2.0 expression language**
 - Best option if server supports JSP 2.0
- **Use Struts `bean:write` tag**
 - Most common approach
 - `bean:write` automatically filters special HTML characters
 - `<` replaced by `<`;
 - `>` replaced by `>`;
 - Etc.

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 6: Display Results in JSP

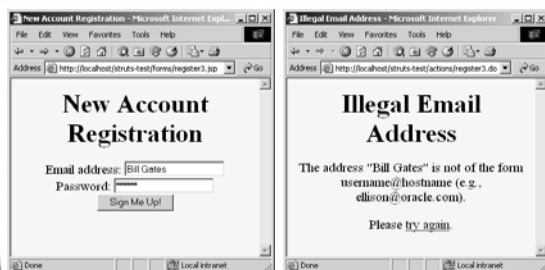
D:\Java\Struts1.1\struts-test3\WEB-INF\results\bad-address3.jsp

```
<!DOCTYPE ...>
<HTML>
<HEAD><TITLE>Illegal Email Address</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<CENTER>
<H1>Illegal Email Address</H1>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
The address
"<bean:write name="userFormBean" property="email"/>"
is not of the form username@hostname (e.g.,
<bean:write name="suggestionBean" property="email"/>).
<P>
Please <A HREF=".."forms/register3.jsp">try again</A>.
</CENTER>
</BODY></HTML>
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 6: Display Results in JSP



Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 6: Display Results in JSP

D:\Java\Struts1.1\struts-test3\WEB-INF\results\bad-password3.jsp

```
<!DOCTYPE HTML ...>
<HTML>
<HEAD><TITLE>Illegal Password</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<CENTER>
<H1>Illegal Password</H1>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
The password
"<bean:write name="userFormBean" property="password"/>"
is too short; it must contain at least six characters.
Here is a possible password:
<bean:write name="suggestionBean" property="password"/>.
<P>
Please <A HREF=".."forms/register2.jsp">try again</A>.
</CENTER>
</BODY></HTML>
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 6: Display Results in JSP



Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 6: Display Results in JSP

```
D:\Java\Struts1.1\struts-test3\WEB-INF\results\result3.jsp
<!DOCTYPE ...>
<HTML>
<HEAD><TITLE>Success</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<CENTER>
<H1>You have registered successfully.</H1>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<UL>
<LI>Email Address:
  <bean:write name="userFormBean" property="email"/>
<LI>Password:
  <bean:write name="userFormBean" property="password"/>
</UL>
</BODY></HTML>
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Step 6: Display Results in JSP



Apache Struts Lecture 1: Intro

www.coreservlets.com

Behind the Scenes: What struts-blank provides

- JAR files already installed in WEB-INF/lib
- TLD files already installed in WEB-INF
 - Not in WEB-INF/tlds, however
- WEB-INF/web.xml with predefined mapping of *.do to Struts controller servlet
- Properties file with standard validator error messages
 - However, properties file name does not match the message-resources name in struts-config.xml.

Apache Struts Lecture 1: Intro

www.coreservlets.com

Behind the Scenes: Using Struts in JHU webdev accounts

- You cannot easily use struts-blank
 - You already have a Web application
 - You cannot make an additional one
 - You already have a web.xml file with various entries in it
- So, you need to insert the relevant pieces of struts-blank in your Web app
 - This is much more work than starting with struts-blank
 - Not usually necessary in real-life applications

Apache Struts Lecture 1: Intro

www.coreservlets.com

Behind the Scenes: Adding Struts to an Existing Web App

- Copy *struts_install_dir/lib/*.*tld*
 - Place in the WEB-INF directory of your Web application
- Copy *struts_install_dir/lib/*.jar*
 - Place in the WEB-INF/lib directory of your Web application
 - Also include struts.jar in your development CLASSPATH
- Modify WEB-INF/web.xml
 - Use servlet and servlet-mapping entries so that URLs that end in *.do are mapped to org.apache.struts.action.ActionServlet
- Create struts-config.xml
 - Define mappings for specific URLs to specific actions

Apache Struts Lecture 1: Intro

www.coreservlets.com

Behind the Scenes: web.xml

```
...
<web-app>
  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>
      org.apache.struts.action.ActionServlet
    </servlet-class>
    ...
  </servlet>
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>
  ...
</web-app>
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Other Struts Capabilities: struts-config.xml

- **<data-sources>**
 - Configure database/JDBC info for easy access
- **<global-forwards>**
 - Addresses associated with common situations, so each action need not specify a specific address
- **<exception> (within <action>)**
 - What to do if uncaught exception is thrown
- **validate="true" (within <action>)**
 - Trigger automatic validation of form beans
- **<messageResources>**
 - Designate the location of a properties file. Properties in that file can then be output via <bean:message>.

Apache Struts Lecture 1: Intro

www.coreservlets.com

Other Struts Capabilities: Form Beans

- **Filling in of form bean can be spread across multiple pages**
 - Maybe it should have been called "user data bean" instead of "form bean"
- **Bean can do automatic validation triggered by validate="true" in the action element.**
 - Results in a call to the validate method of the ActionForm subclass:

```
public ActionErrors validate(ActionMapping mapping,
    HttpServletRequest request) {
```

Apache Struts Lecture 1: Intro

www.coreservlets.com

Other Struts Capabilities: Custom Tags

- **HTML tags**
 - Can associate bean with form
 - Can specify JavaScript code for client-side validation
- **Bean tags**
 - Nested properties
 - property="foo.bar.baz" means to call currentBean.getFoo().getBar().getBaz()
 - <bean:message> outputs values listed in properties file
 - <bean:define> replaces jsp:useBean
 - <bean:cookie>, <bean:parameter>, <bean:header>
- **Logic tags**
 - Looping
 - Matching, comparison, and conditional evaluation
 - JSTL usually preferred now

Apache Struts Lecture 1: Intro

www.coreservlets.com

Related Struts Capabilities: Tiles

- **Define layout template**



- **Dynamically insert content into cells**
- **Change content without editing JSP code**
- **Released with Struts1.1 but is entirely independent of the Struts framework**

Apache Struts Lecture 1: Intro

www.coreservlets.com

Summary

- **Create form that invokes xxx.do**
 - Regular FORM or html:form
- **Create form bean to be populated by the form submission**
 - Extend ActionForm class
- **Create other data beans**
- **Create subclass of Action to handle requests**
- **Use struts-config.xml configuration file to:**
 - Designate that the Action class handle requests for xxx.do
 - Specify URLs that apply in various situations
 - Declare any form beans that are being used
- **Display results using Struts tags**
 - Especially html:text and bean:write

Apache Struts Lecture 1: Intro

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