© 2010 Marty Hall

Jakarta Struts: An MVC Framework

Overview, Installation, and Setup

Customized Java EE Training: http://courses.coreservlets.com/
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

© 2010 Marty Hall

For live Struts training, please see training courses at http://courses.coreservlets.com/.

Taught by the author of *Core Servlets and JSP*, *More Servlets and JSP*, and 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 6, servlets/JSP (intermediate and advanced), Struts, JSF 1.*x*, JSF 2.0, Ajax, GWT 2.0 (with GXT), custom mix of topics
  - Ajax courses can concentrate on 1 library (jQuery, Prototype/Scriptaculous, Ext-JS, Dojo, Google Closure) or survey several
- Courses developed and taught by coreservlets.com experts (edited by Marty)
  - Spring, Hibernate/JPA, EJB3, SOAP-based and RESTful Web Services, Ruby/Rails

Contact hall@coreservlets.com for details

Agenda

- **Understanding Struts**
  - Different views of Struts
  - Advantages of Struts
  - Disadvantages of Struts
- **Setting Up Struts**
  - Downloading and configuring Struts
  - Testing Struts
  - Setting up Struts applications
  - Accessing Struts documentation
  - Adding Struts to an existing Web application

© 2010 Marty Hall

Overview

Customized Java EE Training: http://courses.coreservlets.com/
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

What is Apache Struts?

- **An MVC Framework?**
  - Struts provides a unified framework for deploying servlet and JSP applications that use the MVC architecture.
- **A Collection of Utilities?**
  - Struts provides utility classes to handle many of the most common tasks in Web application development
- **A Set of JSP Custom Tag Libraries?**
  - Struts provides custom tag libraries for outputting bean properties, generating HTML forms, iterating over various types of data structures, and conditionally outputting HTML.
- **But which is the <u>proper</u> way to view Struts?**
  - The answer depends on what you are going to use it for, but the MVC framework is the most common way of looking at Struts.

Advantages of Struts (vs. MVC Using RequestDispatcher)

- **Centralized file-based configuration**
  - Rather then hard-coding information into Java programs, many Struts values are represented in XML or property files. This loose coupling means that many changes can be made without modifying or recompiling Java code, and that wholesale changes can be made by editing a single file. This approach also lets Java and Web developers focus on their specific tasks (implementing business logic, presenting certain values to clients, etc.) without needing to know about the overall system layout.
- **Form beans**
  - In JSP, you can use property="*" with jsp:setProperty to automatically populate a JavaBean component based on incoming request parameters. Apache Struts extends this capability to Java code and adds in several useful utilities, all of which serve to greatly simplify the processing of request parameters.
- **Bean tags**
  - Apache Struts provides a set of custom JSP tags (bean:write, in particular) that let you easily output the properties of JavaBeans components. Basically, these are concise and powerful variations of the standard jsp:useBean and jsp:getProperty tags.

Advantages of Struts (vs. Standard MVC), Continued

- **HTML tags**
  - Apache Struts provides a set of custom JSP tags to create HTML forms that are associated with JavaBeans components. This bean/form association serves two useful purposes:
    - It lets you get initial form-field values from Java objects.
    - It lets you redisplay forms with some or all previously entered values intact.
- **Form field validation**
  - Apache Struts has builtin capabilities for checking that form values are in the required format. If values are missing or in an improper format, the form can be automatically redisplayed with error messages and with the previously entered values maintained.
    - This validation can be performed on the server (in Java), or both on the server and on the client (in JavaScript).
- **Consistent approach**
  - Struts encourages consistent use of MVC throughout your app.

Disadvantages of Struts (vs. MVC with RequestDispatcher)

- **Bigger learning curve**
  - To use MVC with the standard RequestDispatcher, you need to be comfortable with the standard JSP and servlet APIs. To use MVC with Struts, you have to be comfortable with the standard JSP and servlet APIs and a large and elaborate framework that is almost equal in size to the core system. This drawback is especially significant with smaller projects, near-term deadlines, and less experienced developers; you could spend as much time learning Struts as building your actual system.
- **Worse documentation**
  - Compared to the standard servlet and JSP APIs, Struts has fewer online resources, and many first-time users find the online Apache documentation confusing and poorly organized. There are also fewer books on Apache Struts than on standard servlets and JSP.

Disadvantages of Struts (vs. Standard MVC), Continued

- **Less transparent**
  - With Struts applications, there is a lot more going on behind the scenes than with normal Java-based Web applications. As a result, Struts applications are:
    - Harder to understand
    - Harder to benchmark and optimize
- **Rigid approach**
  - The flip side of the benefit that Struts encourages a consistent approach to MVC is that Struts makes it difficult (but by no means impossible) to use other approaches.

Advantages of JSF 1.x (vs. Struts 1.x)

- **Custom components**
  - JSF makes it relatively easy to combine complex GUIs into a single manageable component; Struts does not
- **Better support for Ajax**
  - Several third-party component libraries have extensive Ajax support (Apache Tomahawk, JBoss Ajax4jsf/RichFaces, Oracle ADF, IceFaces). Struts doesn't have real component libraries (see above).
- **Support for other display technologies**
  - JSF is not limited to HTML and HTTP; Struts is
- **Access to beans by name**
  - JSF lets you assign names to beans, then you refer to them by name in the forms. Struts has a complex process with several levels of indirection.

Advantages of JSF (vs. Struts), Continued

- **Expression language**
  - The JSF expression language is more concise and powerful than the Struts bean:write tag.
    - This is less advantageous if using JSP 2.0 anyhow.
- **Simpler controller and bean definitions**
  - JSF does not require your controller and bean classes to extend any particular parent class (e.g., Action) or use any particular method (e.g., execute). Struts does.
- **Simpler config file and overall structure**
  - The faces-config.xml file is much easier to use than is the struts-config.xml file. In general, JSF is simpler.
- **More powerful potential tool support**
  - The orientation around GUI controls and their handlers opens possibility of simple to use, drag-and-drop IDEs

Disadvantages of JSF (vs. Struts)

- **Established base and industry momentum**
  - Struts got there first and has more deployed applications
    - **1/2010 search at dice.com and monster.com**
      - "struts": 1276 jobs (dice.com), 596 jobs (monster.com)
      - "jsf": 480 (dice), 615 (monster) but some listings use "JavaServer Faces"

Disadvantages of JSF (vs. Struts), Continued

- **Support for other display technologies**
  - JSF is not limited to HTML and HTTP; Struts is
    - Hey! Didn't I say this was an *advantage* of JSF?
- **Confusion vs. file names**
  - The actual pages used in JSF end in *.jsp* (JSF 1.x) or *.xhtml* (JSF 2.0). But the URLs used end in *.faces* or *.jsf*. This causes many problems; in particular, in JSF:
    - Cannot browse directories and click on links
    - Hard to protect raw JSP pages from access
    - Hard to refer to non-faces pages in faces-config.xml
- **Self-submit approach**
  - With Struts, the form (blah.*jsp*) and the handler (blah.*do*)have different URLs; with JSF they are the same.

Disadvantages of JSF (vs. Struts), Continued

- **No builtin equivalent to Tiles in JSF 1.x**
  - Struts includes powerful page layout facility; JSF 1.x does not
    - Facelets is not officially part of JSF 1.x (but is in JSF 2.0)
- **Much weaker automatic validation**
  - Struts comes with validators for email address, credit card numbers, regular expressions, and more. JSF only comes with validators for missing values, length of input, and numbers in a given range.
    - But MyFaces and other third-party component libraries have several powerful validators
- **Lack of client-side validation**
  - Struts supports JavaScript-based form-field validation; JSF does not
- **Worse installation**
  - Struts has the struts-blank application to use for a starting point; JSF has nothing similar

© 2010 Marty Hall

Getting Started

Customized Java EE Training: http://courses.coreservlets.com/
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Downloading and Configuring Struts

- **Download the Struts zip file**
  - Start at http://jakarta.apache.org/site/binindex.cgi, or follow the link from http://jakarta.apache.org/struts/.
- **Unzip into a directory of your choice**
  - For example, unzip into C:\jakarta-struts-1.2.4.
- **Add components to new projects**
  - Required
    - web.xml with special settings, several JAR files, several Struts-specific XML files, several TLD files
  - Obtaining files
    - All can be found in the struts-blank project found in the installation folder. As discussed later, you can make a new Web app and copy the individual pieces or copy and rename an entire existing app

Accessing Struts Documentation

- **Main API (JavaDoc)**
  - http://struts.apache.org/struts-action/apidocs/
- **User's Guide**
  - http://struts.apache.org/struts-action/userGuide/
- **Alternative for main API: read a local copy**
  - This is fastest, but the documentation can get a bit out of date. To read a local copy, install the struts-documentation.war Web app. Copy *struts_install_dir*/webapps/struts-documentation.war to *tomcat_install_dir*/webapps, restart server, and use the URL **http://localhost/struts-documentation/**.

Testing Struts in an IDE

- **Install the struts-blank project**
  - See http://courses.coreservlets.com/Course-Materials/struts.html
- **Deploy the project (Eclipse)**
  - Deploy project
    - Select "Server" tab at bottom, R-click on Tomcat, Choose "Add and Remove", Choose project, Click "Finish"
  - Start Tomcat
    - Select "Server" tab at bottom, R-click on Tomcat, Choose "Start"
    - Open browser and enter http://localhost/struts-blank/
    - Will redirect to http://localhost/struts-blank/Welcome.do
- **Deploy the project (JDeveloper)**
  - Right-click on WebContent/index.jsp and select "Run"
  - URL will pop up in browser automatically
  - Will redirect to http://localhost/struts-blank/Welcome.do

Testing Struts with Manual Deployment

- **Install struts-blank.war.**
  - Install the Web application from *struts_install_dir*/webapps/struts-blank.war on your server. For example, with Apache Tomcat, copy *struts_install_dir*/webapps/struts-blank.war to *tomcat_install_dir*/webapps/.
- **Start or restart the server.**
  - Most servers only recognize new Web apps when the server is started.
- **Access http://localhost/struts-blank/.**
  - Will redirect to http://localhost/struts-blank/Welcome.do
  - This URL assumes you are running the server on your desktop and are using port 80. In general, access http://*hostname*:*port*/struts-blank/.

Testing Struts (Results)

Development Process with Eclipse

- **Option 1: copy existing Struts project**
  - Copy and rename struts-blank or another Struts project
  - Due to Eclipse bug, this requires you to manually edit a .component file
    - Still, this is generally easier than pasting all of the necessary Struts components into a new project
- **Option 2: make project from scratch**
  - Make new Dynamic Web project
  - Copy all the required pieces from struts-blank or another Struts project

Developing in Eclipse by Copying "Blank" Project

- **Why copy?**
  - Creating new Struts apps in Eclipse is cumbersome
    - There are many files to copy from struts-blank
- **Issue**
  - Cutting/pasting Web app in Eclipse does not change the context path and internal project deploy name
  - Context path can be changed by R-clicking project, then Properties → Web Project Settings
  - But deployed name is not accessible from within Eclipse!
- **Solution**
  - Go to file system, edit .settings/org.eclipse.wst.common.component
  - Change *two* instances of old project name to new one
  - R-click on project and choose Refresh

Copying struts-blank: Example

- **First, copy project**
  - R-click on struts-blank, choose "Copy"
  - R-click in Project Explorer window, choose "Paste"
    - E.g., name it my-struts-project
- **Next, edit .component file**
  - Navigate to *Eclipse-workspace/projectName/*.settings

Copying struts-blank: Example (Continued)

- **.component file: before**

```
<?xml version="1.0" encoding="UTF-8"?>
<project-modules id="moduleCoreId" project-version="1.5.0">
    <wb-module deploy-name="struts-blank">
        <wb-resource deploy-path="/" source-path="/WebContent"/>
        <wb-resource deploy-path="/WEB-INF/classes" source-path="/src"/>
        <property name="context-root" value="struts-blank"/>
        <property name="java-output-path"/>
    </wb-module>
</project-modules>
```

- **.component file: after**

```
<?xml version="1.0" encoding="UTF-8"?>
<project-modules id="moduleCoreId" project-version="1.5.0">
    <wb-module deploy-name="my-struts-project">
        <wb-resource deploy-path="/" source-path="/WebContent"/>
        <wb-resource deploy-path="/WEB-INF/classes" source-path="/src"/>
        <property name="context-root" value="my-struts-project"/>
        <property name="java-output-path"/>
    </wb-module>
</project-modules>
```

Copying Web Applications in JDeveloper

1. **In Windows, find struts-blank folder**
   - E.g., …/mywork/NBME-Samples/struts-blank
2. **Copy and rename the folder**
   - E.g., to my-struts-app
3. **Open new folder and rename struts-blank.jpr**
   - E.g., to my-struts-app.jpr
4. **In JDeveloper, File → Open, select my-struts-app.jpr**
5. **In new project, R-click, Project Properties → J2EE Application → J2EE Web Context Root**

Adding Struts to an Existing Web Application

- **Copy files into your Web app**
  - Copy JAR files from struts-blank/WEB-INF/lib to your_web_app/WEB-INF/lib.
  - Copy TLD files from struts-blank/WEB-INF to your_web_app/WEB-INF.
  - Copy struts-config.xml from struts-blank/WEB-INF to your_web_app/WEB-INF.
  - Copy the application properties file from struts-blank/WEB-INF/classes/resources to your_web_app/WEB-INF/classes/resources.
  - If you plan on using the automatic validator, copy validation.xml and validator-rules.xml from struts-blank/WEB-INF to your_web_app/WEB-INF.
  - If you plan on using Tiles, copy struts-tiles.xml from struts-blank/WEB-INF to your_web_app/WEB-INF.
- **Update web.xml**
  - Copy declarations out of struts-blank/WEB-INF/web.xml into your_web_app/WEB-INF/web.xml. Most importantly, copy the servlet and servlet-mapping entries that map *.do to org.apache.struts.action.ActionServlet.

© 2010 Marty Hall

Sites that Use Struts

Customized Java EE Training: http://courses.coreservlets.com/
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Some Struts Sites: Travel

Some Struts Sites: Government

Some Struts Sites: Commercial

Struts is Mainstream: Conservatives *and* Liberals Use It

Summary

- **Struts is an MVC framework**
  - Also a collection of utilities and custom tag libraries
- **Struts has significant advantages, but adds significant complexity**
  - Compare the use of Struts to the use of normal MVC (with the JSP 2.0 expression language)
  - Also consider other frameworks, especially JSF 2.0
- **Struts is pretty widely used**
- **Using Struts**
  - Test by deploying struts-blank.war
  - Develop by copying struts-blank or other Struts project
    - Don't make app from scratch; too many changes needed

© 2010 Marty Hall

Questions?

Customized Java EE Training: http://courses.coreservlets.com/
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.