

## Index of Contents

- 1) *What is Struts?*
- 2) *What is a framework?*
- 3) *What are Model-1 and Model-2 architectures?*
- 4) *Core Struts classes as they relate to MVC*
- 5) *Discussing ActionForward, ActionForm, ActionServlet, Action, ActionMapping classes*
- 6) *Discussing struts configuration file*
- 7) *Exploring struts with the Logon Application*
- 8) *Dissecting the Logon Application*
- 9) *Understanding Welcome.jsp, Login.jsp, Syllabus.jsp screens*
- 10) *Description of elements used in Welcome.jsp*
- 11) *Description of elements used in Login.jsp*
- 12) *The LogonForm source*
- 13) *The LogonAction source*
- 14) *The LogOffAction source*

1

### What is Struts ?

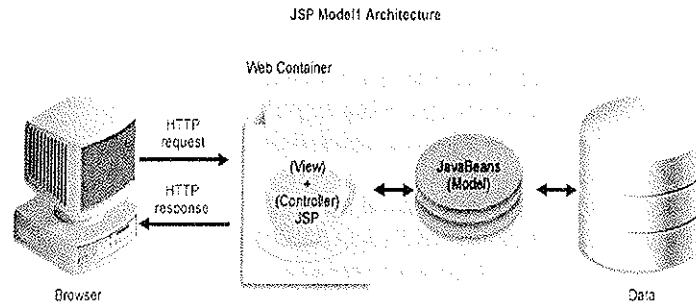
- Struts is open source software framework that helps developers build web applications quickly and easily. Struts relies on standard technologies – such as JavaBeans, Java Servlets and JavaServer Pages.

### What is a framework?

- A framework is a reusable, semi-complete application that can be specialized to produce custom applications.

2

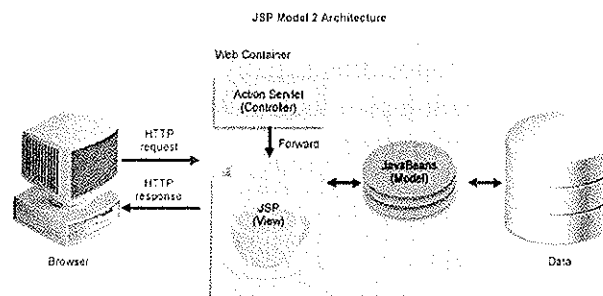
## What is Model 1 architecture?



In Model 1 architecture, the JSP page not only contains the display elements to output HTML, but is also responsible for extracting HTTP request parameters, call the business logic (implemented in JavaBeans, if not directly in the JSP), and handle the HTTP session. Although Model 1 is suitable for simple applications, this architecture usually leads to a significant amount of scripts embedded and makes maintenance difficult.

3

## What is Model 2 / MVC architecture?



Model 2 architecture uses Servlets and JSP pages together in the same application where, Servlets handle the data access and navigation flow, while JSP pages handle the presentation. It lets Java engineers and HTML developers work on their own part of the application. A change in one part of the model 2 application does not mandate a change to another part of the application.

4

### What is Struts architecture based on?

- Struts implements the Model 2 architecture by providing a controller Servlet that can be used to manage the flow between JSP pages and other presentation layer devices. Struts implements the MVC pattern through the use of ActionForwards and ActionMappings to keep control-flow decisions out of the presentation layer.

5

### Core Struts classes as they relate to MVC

Class	Description
ActionForward	A user gesture or view selection
ActionForm	The data for a state change
ActionServlet	The part of the controller that receives user gestures and state changes and issues view selection
Action classes	The part of the controller that interacts with the model to execute a state change or query and advises the ActionServlet of the next view to select.
ActionMapping	The state change event

#### ***ActionForward***

This class can be used to store the path to a page under a logical name. The Servlet uses the path stored in the ActionForward object to call the page and complete the response.

6

### ***ActionForm***

This class is designed to handle input from an HTML form, validate the input, and redisplay the form to the user for correction (if required), along with any corresponding prompts or messages. ActionForms are just JavaBeans.

### ***ActionServlet***

When the ActionServlet receives a request from the container, it uses the URI to determine which Action it will use to handle the request. The ActionServlet bundles HTML input into a JavaBean. The input beans are subclasses of the Struts ActionForm class.

### ***Action***

An action can validate input and access the business layer to retrieve information from database and other data services.

7

### ***ActionMapping***

Struts bundles all objects (*ActionServlet*, *Action*, *ActionForward*) into an *ActionMapping* object. Each *ActionMapping* is related to a specific path. When that path is requested, the servlet retrieves the *ActionMapping* object. The mapping tells the servlet which Action, ActionForms, and ActionForwards to use. All of these details are declared in the *struts-config.xml* file

### ***Struts-config.xml***

Struts uses a configuration file to define several things about your application, including the logical names for hyperlinks. This is an XML document that Struts reads at startup and uses it to create a database of objects. Various Struts components refer to this database to provide the frameworks's services.

8

### ActionMapping settings

Property	Property
path	A unique identifier for this mapping. It is included in the web address, as in <a href="http://localhost:8080/logon/LogonSubmit.do">http://localhost:8080/logon/LogonSubmit.do</a>
type	The Action object to call when the path is requested.
name	The JavaBean helper (ActionForm) to use with an HTML form.
scope	A property that specifies whether to store the helper in the request or the session
validate	A property that specifies whether to call the standard validate method on the form bean helper (specified by <i>name</i> ) before calling the Action object (specified by <i>type</i> )
input	A property that specifies where to send control if the <i>validate</i> method returns false.

9

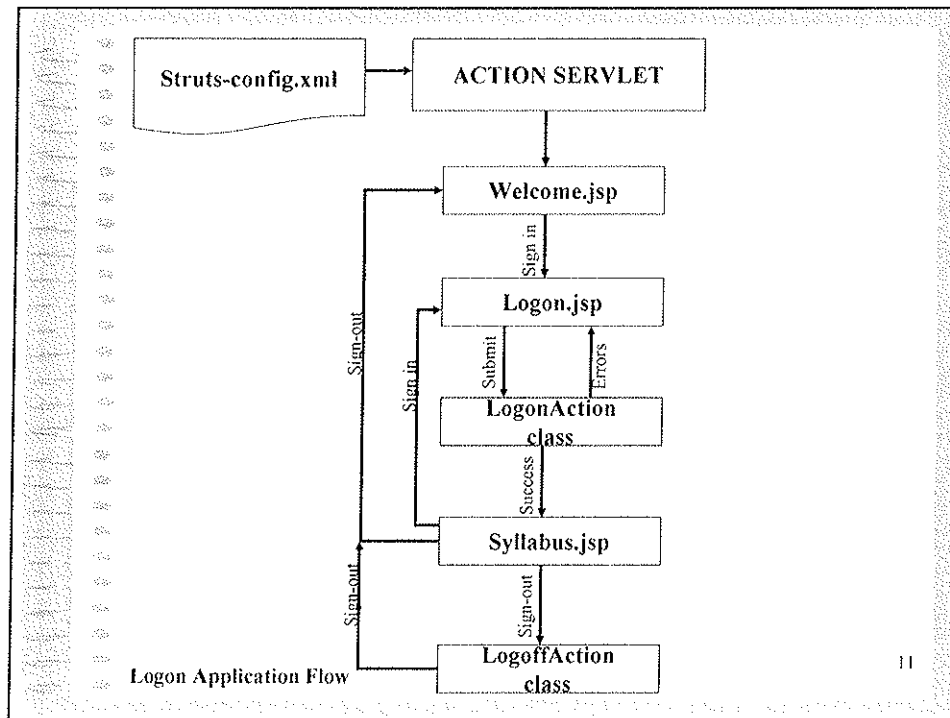
### Struts configuration file

File	Purpose
ApplicationResources.properties	Stores localized messages and labels so that your application can be internationalized
Struts-config.xml	Stores the default configuration for the controller objects, which includes user gestures, state changes, and state queries supported by your mode.

hey, what tools are required  
for working on Struts ?

- JDK 1.4 or higher
- Tomcat 4.x or higher or any web/application server (Websphere, Weblogic etc.)
- Apache Struts

10



## Dissecting the Logon application

The logon application allows the user to log into the application. Once the user has logged in, the pages change to reflect that the user is now authorized. Typically this is the first step in a larger application but for our purposes, just logging in a user and displaying the syllabus page is enough to show you how a Struts application actually works.

### Screens used by the Logon application

Screen	Purpose
Welcome.jsp	Greets visitor and offers links into the application.
Logon.jsp	<ul style="list-style-type: none"> <li>- Allows input of user name and password and submits it.</li> <li>- If the form is submitted without entering the user name or password, appropriate message(s) will be displayed.</li> <li>- It tells the user everything that is missing all at once.</li> </ul> <p>When user submits only one thing, it reminds the user only about the other thing.</p> <ul style="list-style-type: none"> <li>- If valid user name and password are entered, the next screen is displayed.</li> </ul>
Syllabus.jsp	<p>Displays course syllabus after successful login.</p> <ul style="list-style-type: none"> <li>- Offers links to sign-on and sign-out.</li> <li>- If sign-out link is clicked; user is returned to the original welcome screen (the one with just the sign-in link).</li> <li>- If sign-in link is clicked; user is returned to the logon screen to enter user name and password.</li> </ul>

12

## Welcome.jsp

Welcome World!

- [Sign in](#)

Designed by Jay & Venkat  
- [www.JTutor.net](http://www.JTutor.net)

Entering <http://localhost:8080/login> in IE will bring up the Welcome.jsp page shown above.

13

## Login.jsp

Username:

Password :

**Submit** | **Reset**

Screen 1

You will see this screen, if [Sign in](#) link is clicked on the Welcome.jsp page.

**Validation Error**

You must correct the following error(s) before proceeding:

- Username is required
- Password is required

Username :

Password :

**Submit** | **Reset**

Screen 2

You'll see this screen, if Submit button is clicked without entering user name and password.

14

## Syllabus.jsp

Welcome J2EE!

### Core Java Programming

Introduction to Java and language fundamentals  
Declaration and Access Control  
Operators and Assignments  
Flow control and Exceptions  
Object orientation, Overloading and Overriding, Constructors and Return types  
java.lang \* - The Math class, String and Wrappers  
Collections  
Threads

- [Sign in](#)
- [Sign out](#)

Designed by Jay & Venkat  
- [www.JTutor.net](http://www.JTutor.net)

Above page will be displayed after successfully logging into the application.

15

## Description of JSP elements used in Welcome.jsp

### 1. taglib

These are JSP equivalents to import statements and make the tag extensions available to the rest of the JSP page.

```
<%@ taglib uri="/tags/struts-html" prefix="html"%>  
<%@ taglib uri="/tags/struts-logic" prefix="logic"%>
```

→ *html: <link>, etc.*

### 2. logic tag

This segment looks to see if we have stored a "user" bean in the client's session.

If such a bean is present, then the user is welcome by name.

```
<logic:present name="user">  
<H3>Welcome <bean:write name="user" property="username"/>!</H3>  
</logic:present>
```

Conversely, if the user bean is not present, then we use a generic welcome.

```
<logic:notPresent scope="session" name="user">  
<H3>Welcome World!</H3>  
</logic:notPresent>
```

16



### The configuration source for the Welcome screen

Here, Logon is a key that is used to look up the actual path for the hyperlink

```
<forward name="logon" path="/Logon.do"/>
```

### Description of JSP elements used in logon.jsp

```
<html:errors/>
```

You'll notice that if we try to submit the form without entering a logon, an error message is displayed. The above tag is the one that renders the error messages.

```
<html:form action="/LogonSubmit" focus="username">
```

The `<html:form>` tag produces an HTML form for data entry. The action parameter tells the `<html:form>` tag which ActionMapping to use. This is defined in Struts configuration file.

17

This following tag creates standard HTML Submit and Reset buttons.

```
<TD align="right"><html:submit/></TD>
```

```
<TD align="left"><html:reset/></TD>
```

The configuration source for the Logon screen

Element	Description
/LogonSubmit ActionMapping	Encapsulates several details needed when building and submitting an HTML form with the Struts framework
app.LogonForm	Describes properties used by the HTML form
app.LogonAction	Completes the processing of the submitted form.

18

### The LogonForm source

When an HTML form is submitted, the name-value couplets are caught by the Struts controller and applied to an ActionForm. The ActionForm is a JavaBean with properties that correspond to the controls on an HTML form. When they match, the controller sets the property to the value of the corresponding couplet. Extra properties are ignored.

When the mapping is set to `validate=true`, the `validate` method is called after the form is populated from the HTTP request.

The `ActionErrors` object returned by `validate` is another framework class. If `validate` does not return null, then the controller will save the `ActionErrors` object in the request context under a known key. The `<html:errors>` tag knows the key and will render the error messages when they exist, or do nothing when they do not.

The tokens `error.username.required` and `error.password.required` are also keys. They are used to look up the actual messages from the Struts message resource file, `application.properties`.

19

### The LogonAction source

After collecting the data entry into an ActionForm and performing any initial validations, the controller passes the form along to the Action class given by the mapping.

When a request for an Action is sent to the Struts servlet, it invokes the Action by calling its *perform* (or *execute*) method.

The purpose of an Action is to ferry input from the web tier to the business tier where the rest of your application lives. Here we extract the user name and password and pass them to the business tier function to see if they validate.

If logon is no good, the Action posts the error message and routes control back to the input page

If logon is good, then we retain the user's credentials. Here, we store the user's logonForm in their session context as follows.

20

```

HttpSession session = request.getSession();
session.setAttribute(Constants.USER_KEY, form);

```

When everything is done, the perform method returns an ActionForward to the controller (ActionServlet). Here, we send control to the success forward:

```

// return success
return (mapping.findForward(Constants.SUCCESS));
}

```

This is defined in the Struts configuration as:

```
<forward name="success" path="/pages/Welcome.jsp"/>
```

#### **The LogoffAction source**

Once the user logs on, the Syllabus page provides a link for sign-off. It is defined in Syllabus.jsp as

```
<html:link forward="logoff">Sign out</html:link>
```

21

In the Struts configuration file, the logoff forward is defined as:

```
<action path="/Logoff" type="app.LogoffAction">
```

The job of the LogoffAction class is very simple. It removes the user's logonForm object from the session context using the following statement.

```

// Remove user login
session.removeAttribute(Constants.USER_KEY);

```

If we wanted to remove everything from the session that might be stored for the user, we could simply invalidate the session instead using:

```
session.invalidate();
```

As with logon, when the operation completes, we return to the welcome page using:

```

// Return success
return (mapping.findForward(Constants.SUCCESS));

```

22

