Struts framework

Introducción

Qué es Struts

Struts es un framework open-source creado por Craig Mc Clanahan, y luego donado a la Fundación Apache para su mantenimiento. Tiene como objetivo organizar y simplificar la capa de presentación de aplicaciones construidas con JSP, combinando el uso de JSP, Servlets, Tag Libraries y clases básicas de Java.

Su configuración es compleja, pero una vez realizada, la aplicación tiende a crecer de forma organizada, consiguiendo maximizar escalabilidad y minimizar mantenimiento. En su aspecto de distribución, el framework está compuesto por archivos .jar que hay que agregar al proyecto que utilice Struts.

El modelo MVC

Struts está basado en el patrón de diseño MVC. Las siglas representan partes del modelo MVC, y significan:

M Model

V View

C Controller

El patrón está basado en estas tres partes bien definidas, donde cada una cumple con determinadas responsabilidades

Relación entre MVC y Struts

Struts implementa el patrón de diseño MVC, existiendo clases que modelan cada una de las partes, donde cada clase tiene que cumplir únicamente con sus responsabilidades. Las clases que integrarán la aplicación tomarán como base clases pertenecientes al framework.

Downloads

El framework de Struts se puede conseguir en el siguiente link:

http://struts.apache.org/download.cgi

El Controlador (Controller)

Qué es

Es una de las tres partes del patrón MVC, y se encarga de recibir eventos del usuario y llamar a las acciones que resulten necesarias. Tambien determina la vista a presentar.

El ActionServlet como MasterController

El ActionServlet es la clase base de Struts que se encarga de controlar los pedidos del request. En términos generales, recibe pedidos del request del browser y determina los pasos a seguir, que acción realizar (a través de clases del tipo Action). En Struts, el controller está implementado como un Servlet y su configuración se realiza en el deployment descriptor, o archivo web.xml.

La clase Action

La clase Action forma parte del framework de Struts, y tiene como responsabilidad ejecutar las reglas de negocio. En general, esta clase interactúa con la capa de negocios. Contiene un método denominado execute() que es llamado por el MasterController cuando se ejecuta el Action que lo contiene.Para construir un Action hay que tomar como base la clase org.apache.struts.action.Action

El metodo execute()

El método execute() forma parte de la acción (de la clase Action), es un método perteneciente a la clase org.apache.struts.action.Action que se debe redefinir en la clase Action definida por el usuario. El método se ejecuta automáticamente cuando se llama a la acción, y deberia contener la implementación de las reglas de negocio, o llamar a la capa de negocios correspondiente. Ademas debe determinar qué vista debe ser la próxima a utilizarse a través de una instancia de la clase ActionForward, que es el objeto que retorna éste método.

La clase ActionForward

La clase org.apache.struts.action.ActionForward forma parte del framework de Struts y se utiliza para modelar el próximo destino luego de completar la acción correspondiente. El método execute() del Action al finalizar su ejecución debe elegir el ActionForward correspondiente, es decir la proxima vista a visualizar.

El Modelo (Model)

Qué es

Es una de las tres partes del patrón MVC, representa los datos de la aplicación que pueden ser persistentes o no. En el caso de Struts, no representa información persistente sino información enviada por el browser para ser procesada por el Action. Debe notificar a la vista cuando cambia su estado interno.

La clase ActionForm

Representan el modelo dentro del MVC de Struts, donde maneja los datos enviados a través de formularios HTML (GET/POST) y urls. Posee un atributo por cada tag <INPUT> correspondiente al formulario, junto con sus correspondientes setters y getters. Puede manejar datos a nivel request o session, pero no maneja persistencia de datos. Contiene un método validate() utilizado para realizar validaciones del lado del servidor.

Para construir un ActionForm hay que tomar como base la clase org.apache.struts.action.ActionForm

El método validate()

El método validate() se utiliza para realizar las validaciones del formulario según los datos que recibe, por ejemplo campos vacíos o valores que no son numéricos donde se esperan números. El método genera una colección de errores a través de la clase ActionErrors.

La forma de utilizarlo es la siguiente:

```
public ActionErrors validate

(ActionMapping mapping, HttpServletRequest request)

{

// Construye un objeto errores

ActionErrors errors = new ActionErrors();

// Valida que el nombre sea distinto de vacio

if (this.getNombre() == null)

{

errors.add("nombreVacio",

new ActionMessage("login.error.nombrevacio"));

}

// Retorna los errores

return errors;
}
```

Este método se ejecuta antes de la ejecución del Action, con lo cual si existe algún error el Master Controller redirecciona automáticamente a la página que solicitó originalmente la acción.

Los errores generados se pueden visualizar en pantalla utilizando el tag:

<html:errors/>

La vista (View)

Que es

Es otra de las tres partes del patrón de diseño MVC, es la encargada de "dibujar" la interfaz gráfica de usuario a partir del estado del modelo. Si el modelo cambia, deberá reflejar ese cambio gráficamente. El ingreso de datos por parte del usuario deberá ser enviado al controlador.

La lógica de presentación queda construida dentro de la vista, dentro de JSP la vista queda representada por archivos JSP, pero con la particularidad de que las etiquetas pertenecientes a los formularios (por ejemplo las etiquetas FORM, INPUT y SELECT) deben "migrarse" a tags de Struts.

El archivo .jsp

El archivo JSP representa la vista dentro del modelo MVC de Struts, y se construye a partir de archivos HTML y JSP. Resulta necesario utilizar tags JSP de Struts para construir los formularios.

Struts HTML Tags

Para poder construir la vista y que Struts la "interprete" y pueda utilizar el form correspondiente, resulta necesario armar la página JSP utilizando tags de Struts. Estos tags reemplazan a los tags HTML tradicionales, y estan basados en el uso de JSTL, son tags personalizados creados para manipular distintas partes de la vista.

Para importar los tags de Struts es necesario realizar la importación de las tag libraries que seran utilizadas, de igual manera que fue realizado anteriormente con JSTL.

La importacion de tag libraries de Struts se realiza de la siguiente manera:

```
<%@taglib uri="http://jakarta.apache.org/struts/tags-html"

prefix="html" %>

<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean"

prefix="bean" %>
```

Cada componente perteneciente a HTML posee un tag correspondiente en Struts, por ejemplo el tag <INPUT type=text> tiene su correspondiente como https://doi.org/10.1001/journal.com/ https://doi.org/10.1001/journal.com/ https://doi.org/10.1001/journal.com/ https://doi.org/10.1001/journal.com/ https://doi.org/10.1001/journal.com/ https://doi.org///org/10.1001/journal.com/ https://doi.org///org/ https://doi.org/ https://doi

Por ejemplo, la siguiente pagina esta construida con HTML standard:

```
<form method="POST" action="index.jsp">
<input type="hidden" name=accion value=login>
```

```
username: <input type="text" name=username>
<br>><br>>
password: <input type="password" name=password>
<br>
<hr>>
<input type=submit value=Login><br>
</form>
Al migrar los tags del formulario, a sus correspondientes de Struts, la página queda de la siguiente manera:
<%@ taglib uri="http://jakarta.apache.org/struts/tags-bean"
prefix="bean" %>
<%@ taglib uri="http://jakarta.apache.org/struts/tags-html"
prefix="html" %>
<a href="https://www.energeneurone.com/html:form.action="procesarLogin.do" method="POST">
<a href="https://www.nihiden.google.com"/sulue="login"/>
username: <html:text property="username" />
<br>><br>>
password: <a href="massword">html:password property="password"/>
<br>
<hr>>
<a href="https://submit value="Login"/>
</html:form>
```

Configuración de Struts en el deployment descriptor

Introducción

El archivo web.xml representa el descriptor de despliegue de la aplicación. Para poder utilizar Struts dentro de la aplicación, resulta necesario declarar el controlador maestro (el ActionServlet, también llamado Master Controller) dentro de éste archivo, para que tome control de la navegación y realice el procesamiento necesario.

Declaración del ActionServlet

Para configurar Struts dentro de la aplicación, resulta necesario agregar el siguiente código dentro del archivo web.xml:

```
<servlet>
<servlet-name>action</servlet-name>
<servlet-class>org.apache.struts.action.ActionServlet
</servlet-class>
<init-param>
<param-name>config</param-name>
<param-value>/WEB-INF/struts-config.xml</param-value>
</init-param>
</servlet>
```

La clase ActionServlet representa el controlador maestro, y utiliza como parámetro el directorio logico donde se encuentra el archivo de configuracion struts-config.xml. De esta forma, el ActionServlet al contar con el archivo de configuración, posee el control total de la aplicación.

Mapeo del ActionServlet

Adicionalmente, resulta necesario establecer que solicitudes deberá interceptar el ActionServlet. Para esto se deberán declarar las extensiones a capturar por el ActionServlet, que en el standard de Struts estan representadas por cualquier solicitud que termine con *.do.

A continuación se visualizan las líneas necesarias a agregar dentro del descriptor de despliegue:

<servlet-mapping>

```
<servlet-name>action</servlet-name>
<url-pattern>*.do</url-pattern>
</servlet-mapping>
```

Configuración del archivo struts-config.xml

Introducción

El archivo struts-config.xml es donde se asienta la configuración para que Struts a traves del Servlet (Master Controller) pueda controlar la navegacion, presentación y procesamiento de información.

Resulta necesario registrar tanto los ActionForms a utilizar como los Actions, y contempla también el manejo de excepciones. Los archivos con extension .properties utilizados para internacionalizar la aplicacion también deberán quedar registrados aquí.

Los ActionForm

Los ActionForm debe estar declarados dentro del struts-config.xml para su utilización. La registración de un ActionForm se realiza dentro del tag <form-beans>, de la siguiente manera:

```
<form-beans>
<form-bean name="miForm" type="paquete.MiForm"/>
</form-beans>
```

La propiedad name representa el nombre que utilizará como identificador dentro del archivo de configuración, puede entenderse como el nombre del objeto. La propiedad type representa a qué clase pertenece pertenence este ActionForm, y es necesario anteponer el paquete donde esta ubicada. Por cada ActionForm debe haber un tag <form-bean> dentro del archivo de configuración.

Los ActionMappings

Las clases del tipo Action también deben estar registradas dentro del archivo de configuración. La registración se realiza dentro del tag <action>, de la siguiente manera:

<action-mappings>

```
<action input="/login.jsp" name="loginForm"

path="/procesarLogin" type="paquete.MiAction" />
</action-mappings>
```

La propiedad input representa la página JSP que llamó originalmente a este Action, y se utiliza por si existe algún error dentro del método validate() correspondiente al ActionForm, para que Struts sepa donde redireccionar. La propiedad name indica que ActionForm deberá usar este Action, y debe coincidir con el name utilizado en el tag <form-bean>. La propiedad path representa la url con extensión .do que se deberá solicitar para llamar a este Action, y deberá coincidir con el action dentro del tag <html:form action="procesarLogin">>. Por último, la propiedad type indica a que clase pertenece este Action.

Fowards

Los Forwards son páginas referenciadas por un Action, de acuerdo a cierta necesidad. Cuando se procesa el Action, es posible que todo se resuelva de acuerdo a lo esperado o se genere algún error. El Action debe determinar la próxima página JSP a visualizar de acuerdo a su curso de acción. Para esto, es necesario configurarlo dentro del struts-config.xml utilizando tags <forward> contenidos dentro del tag <action> correspondiente. La registración del tag <forward> se realiza de la siguiente manera:

```
<action-mappings>
<action-mappings>
<action input="/login.jsp" name="LoginForm"

path="/procesarLogin" type="paquete.MiAction">
<forward name="exito" path="/exito.jsp" />
<forward name="fracaso" path="/fracaso.jsp" />
</action>
</action-mappings>
```

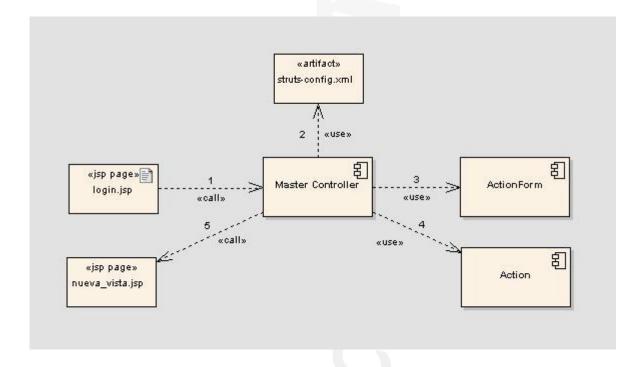
Global Forwards

Los Global Forwards son Forwards que pueden ser referenciados de forma global, es decir que cualquier Action de la aplicación podría necesitarlo. No están necesariamente condicionadas al éxito o fracaso de una acción, y pueden ser invocados desde páginas JSP, por ejemplo desde un link correspondiente a un "logout".

La registración de los Global Forwards se realiza en el struts-config.xml dentro de tags <global-forwards>, de la siguiente manera:

```
<global-forwards>
<forward name="login" path="/login.jsp"/>
<forward name="logout" path="/logout.jsp"/>
</global-forwards>
```

Diagrama de Funcionamiento



La página JSP esta construida con tags de Struts. Cuando el botón submit dentro del formulario es presionado, se envía un pedido al servidor, el cual será capturado por el Servlet de Struts (el Master Controller).

El Master Controller recibe la propiedad action correspondiente al tag https://example.com/html:form del formulario, y la cheque dentro del struts-config.xml para determinar la clase ActionForm y la clase Action a utilizar.

Una vez detectado el ActionForm, el Master Controller lo instancia, lo llena con la información correspondiente al formulario y ejecuta el método validate() para realizar las validaciones correspondientes. Si existen errores, el MasterController redirecciona a la página JSP origen, caso contrario utiliza el Action.

En caso de no haber errores, el Master Controller instancia el Action correspondiente y ejecuta el método execute(), que contiene reglas de negocio (o llamadas a reglas de negocio) junto con referencia a la proxima vista (el proximo JSP a visualizar).

El Master Controller selecciona la proxima vista a visualizar referenciado por el método execute().

Pasos para la construcción de una aplicación con Struts

Paso #1 - Configuración del archivo web.xml:

Registración del ActionServlet con el struts-config.xml como parámetro

Mapeo del ActionServlet con *.do

Paso #2 - Construcción de paquetes para beans y actions

Paso #3 - Construcción de las páginas JSP con tags de Struts

Paso #4 - Construcción del ActionForm correspondiente al formulario, junto con su registración en el struts-config.xml y la implementación del método validate()

Paso #5 - Construcción del Action, junto con su registración en el struts-config.xml y la implementacion del metodo execute()

Paso #6 - Registración de los Forwards correspondientes al Action dentro del struts-config.xml

Los pasos #3 al #6 se repiten cada vez que se construye una nueva página JSP que contiene un formulario.