

Ejercicios sesión 1 - Introducción a Struts

Índice

1 Instalación de la aplicación de ejemplo.....	2
2 Mapear una acción en el struts-config.xml.....	2
3 Implementar un caso de uso completo.....	3
4 Gestionar mensajes de error en las acciones (*).....	3

1. Instalación de la aplicación de ejemplo

Durante las tres sesiones de Struts realizaremos los ejercicios sobre la misma aplicación de ejemplo. Se trata de una sencilla aplicación para almacenar las tareas pendientes (ToDo) de un usuario. Hay que tener en cuenta que es simplemente una aplicación de ejemplo. El objetivo es tener una aplicación para poder "trastear" con Struts más compleja que un "hola mundo" pero no tanto como una aplicación real. Tanto la arquitectura de la aplicación como el interfaz de usuario se deberían mejorar en una implementación más realista.

Los casos de uso de la aplicación son:

- Hacer login
- Ver la lista de tareas pendientes
- Crear una nueva tarea
- Eliminar una tarea existente

Para instalar la aplicación, seguid estos pasos:

1. Bajáos el proyecto de Eclipse e importadlo a vuestro espacio de trabajo
2. **Crear la base de datos:** ejecutad el *target* de Ant denominado *crearBD* que viene en el *build.xml*
3. **Comprobar que Struts funciona:** ejecutad la aplicación y acceded a la URL <http://localhost:8080/StrutsToDo/prueba.do>. Debe aparecer una página confirmando que todo está bien. El formulario para hacer login en la aplicación está en *index.jsp*, pero por el momento no funciona (hasta que lo arreglemos en el ejercicio 1).

2. Mapear una acción en el struts-config.xml

El formulario de login de la página *index.jsp* llama a *login.do*, pero falta mapear esta URL a la acción en el *struts-config.xml*. **Realizar el mapeo** usando la etiqueta `<action>`. La clase asociada será *es.ua.dccia.presentacion.acciones.AccionLogin*. Examinar el código de esta clase para ver los posibles resultados que devuelve, creando un `<forward>` para cada resultado.

Aviso:

Recordad que cualquier modificación del fichero *struts-config.xml* requerirá la recarga de la aplicación en el servidor, ya que Struts no detecta automáticamente los cambios en este fichero.

Una vez hecho el mapeo, comprobar que el login funciona. En la base de datos hay un usuario con login "struts" y password "mola".

3. Implementar un caso de uso completo

Como mejor se ve el funcionamiento de Struts es siguiendo el flujo de ejecución de un caso de uso completo. Para ello vais a implementarlo vosotros y luego probarlo. El caso de uso en cuestión es el de mostrar todos los datos de una tarea. En la página que lista todas las tareas ya hay un enlace que apunta a "verTarea.do" al lado de cada tarea. Falta por implementar:

1. La clase Java con la acción. Se debe llamar `es.ua.dccia.presentacion.acciones.AccionVerTarea`.
 - La acción debe tener tres resultados distintos (*forwards*): "OK", "error" (si se produce algún error por ejemplo con la base de datos) y "noAutorizado" (si el usuario no ha hecho login)
 - Fijaos en la implementación de alguna otra acción (por ejemplo `AccionListarTodas`) para ver cómo se controla el login del usuario.
 - El acceso a la base de datos nos lo da la clase `TareaDAO`. Para obtener una instancia, llamar a `TareaDAO.getInstance()`. Para obtener los datos de una tarea, llamar a `getTarea(id_de_la_tarea, login_del_usuario)`
 - Una vez se obtenga un objeto `Tarea` con todos los datos, colocarlo en el `request` para que se pueda mostrar desde un JSP
2. El JSP que muestre los datos de la tarea. Usad el lenguaje de expresiones de JSP para simplificar la sintaxis, podéis tomar como ejemplo `tareas.jsp`.
3. El mapeo en `struts-config.xml` entre la URL, la acción y el JSP con la vista

4. Gestionar mensajes de error en las acciones (*)

En la acción `AccionLogin` que se os da en el ejemplo se comprueba que login y password son correctos, volviendo al `index.jsp` en caso de que no lo sean, pero no se muestra ningún mensaje de error. Hay que: (consultad los apuntes para ver los detalles de cómo hacerlo)

1. En el fichero `mensajes.properties` (en la carpeta `src`) añadir un mensaje de error para este caso
2. En el código java de la acción Crear un `ActionMessage` asociado al mensaje anterior y guardarlo con `saveErrors` cuando se detecte que login y/o password son incorrectos.
3. En `index.jsp` usar la etiqueta `<html:messages>` para mostrar el error.

