

Ejercicios de MVC en Spring

Índice

1 MVC sin procesamiento de datos de entrada.....	2
2 MVC con procesamiento de datos de entrada.....	2
3 Validación de datos (*).....	2

Continuaremos en esta sesión con la aplicación AmigosSpring, refactorizando la capa de presentación para que pase de estar basada en servlets a usar Spring MVC

1. MVC sin procesamiento de datos de entrada

Cambiar el caso de uso "leer mensaje" para que emplee Spring MVC. Ello implicará:

1. Introducir un mapeo de las URL *.mvc para que se redirijan al `DispatcherServlet` de Struts
2. Cambiar el servlet `LeerMensajeServlet` por un *controller* de Spring
3. Crear un XML para configurar la parte MVC y añadir este controller como un bean de Spring, además de un `ViewResolver`.

2. MVC con procesamiento de datos de entrada

Cambiar el caso de uso "buscar usuarios" para que emplee Spring MVC. Ello implicará:

1. Definir un `Command` que pueda capturar los datos que se introducen en el formulario de búsqueda
2. Cambiar el servlet `BuscarUsuariosServlet` por un *controller* de Spring
3. Configurar este controller como un bean de Spring en el XML correspondiente.

3. Validación de datos (*)

Implementar validación de datos para el caso de uso "buscar usuarios". En concreto, verificar que tanto la edad mínima como la máxima buscadas contienen algún valor, que la máxima es efectivamente un número mayor que la mínima, y que ninguna es menor de 18. Acordaos de:

1. Crear una clase que implemente `Validator` y que contenga el código de validación
2. Configurarlo adecuadamente en el XML (como un bean y además como una propiedad "validator" del controller)
3. Crear el fichero `.resources` con los mensajes y configurar su nombre en el XML de definición de beans.
4. Añadir las etiquetas de Spring a la página `.jsp` para que se muestren tanto los mensajes de error como los valores introducidos por el usuario en los campos, si se produce un error de validación.

