# Ejercicios de validación e internacionalización

## Índice

1 Conversión y formateo de datos (0.5 puntos)	2
2 Validación (1.5 puntos)	. 2
3 Internacionalización (1 punto)	3

En las plantillas de la sesión tienes un proyecto ya configurado con Spring. La aplicación es para la gestión de una agencia de alquiler de coches. Se pueden listar los coches en alquiler, insertar nuevos o editar los existentes. Para simplificar no hay autentificación de usuarios.

En la carpeta "database" del proyecto tienes el script SQL con la base de datos mySQL. Ejecútalo como lo hacemos habitualmente para crear la base de datos. Comprueba que el listado de coches aparece correctamente accediendo a "listar.do".

Si quieres probar a dar de alta un coche o modificar los datos de uno existente debes introducir la fecha en un formato como 01-ene-2010 o 05-oct-2011. En el segundo ejercicio veremos cómo cambiar el formato de las fechas.

### 1. Conversión y formateo de datos (0.5 puntos)

Vamos a añadir la funcionalidad de que se puedan listar solo los vehículos matriculados con posterioridad a una fecha dada. Modifica el método listar de la clase es.ua.jtech.spring.mvc.CocheController para que admita un nuevo parámetro HTTP "fecha", opcional. Si fecha es null, listará todos los coches. Si no lo es, listará solo los matriculados con posterioridad a esa fecha. Tendrás que:

- 1. Añadirle un parámetro al método listar y vincularlo al parámetro HTTP "fecha". Recuerda que esto se hacía con "@RequestParam". El parámetro HTTP debe ser opcional.
- 2. Anotar convenientemente el parámetro para que se haga la conversión de String a Date. El formato de entrada será de dos dígitos para día,mes y año separados por barras ("/"). Al ser un formato personalizado tendrás que usar un "pattern"
- 3. Modificar el código del método para que si la fecha es null use el "listar()" del DAO y en caso contrario llame al "listarPosterioresA(fecha)".
- 4. Conprueba que todo funciona pasando el parámetro "manualmente" en la URL: "http://localhost:8080/AlquilerCoches/listar.do?**fecha=01/01/09**"

### 2. Validación (1.5 puntos)

Modifica la aplicación para validar los datos del coche cuando se dé de alta o se edite. Tendrás que:

- Poner las anotaciones JSR303 pertinentes en los campos de la clase org.especialistajee.spring.modelo.Coche. Debes validar que la matrícula tiene 7 caracteres de largo, que ni el kilometraje del coche ni el precio pueden ser números negativos, y que la fecha de matriculación del coche debe ser ya pasada.
- Para que Spring convierta la fecha que escribas en el formulario, que será un String, a un Date, debes anotar el campo "fechaMatriculacion" con DateTimeFormat. Usa el mismo formato con el que aparecen los coches en el listado (dos dígitos para día, mes y año y separados por barras).

- Modifica la cabecera de los métodos procesarCrear y procesarEditar de la clase org.especialistajee.spring.mvc.CocheController para asegurar que Spring valida los objetos Coche que se le pasan.
- Modifica el fichero mensajes.properties de "src/main/resources", añadiendo las claves necesarias para que se muestren los mensajes de error de validación. Puedes probar a dar de alta algún coche con datos no válidos para comprobar que se vuelve al mismo formulario. Como el log está configurado para ello, verás aparecer los errores de validación en la consola de Tomcat de Eclipse. Pero por supuesto no en el JSP, cosa que vamos a solucionar ahora. Cuando pruebes el funcionamiento, ten en cuenta que estás controlando los errores de validación, pero para los de conversión de tipos tendrías que crear los mensajes del tipo "typeMismatch" en el .properties, como hicimos en la sesión 3.
- Introduce en los JSP "crear.jsp" y "editar.jsp" (están en src/main/webapp/WEB-INF/views) *tags* de Spring para mostrar los errores al usuario. Al lado de cada campo debes meter una etiqueta form:errors, por ejemplo para la matrícula:

<form:errors path="matricula"/>

#### 3. Internacionalización (1 punto)

Prepara la aplicación para que se vea en inglés y español. El localeChanger ya está configurado en el XML de Spring. Guardará el locale en una cookie del navegador. Tendrás que:

- Añadir un nuevo *locale* para inglés al mensajes.properties. Por tanto, tendrás que cambiar el nombre actual por mensajes\_es.properties y crear una copia del archivo como mensajes\_en.properties. Edita esta última y traduce los mensajes.
- Modificar los JSP para que muestren los textos con la etiqueta message de Spring en lugar de estar escritos directamente en el código. Traduce al menos la página del listado de coches al inglés y al español, y añade un enlace en dicha página para poder cambiar el idioma. El enlace debe apuntar a "listar.do?locale=en\_us" para inglés de USA y lo mismo con "locale=es\_es" para español. Si no pones el país en el locale, hay cosas que serán ambiguas, como la moneda.
- Formatea el kilometraje del coche, la fecha de matriculación y la moneda para que se adapte al idioma que se está usando. Para la fecha usa un formato corto, sin hora. Evidentemente el valor de la moneda no será correcto, ya que además de cambiar el símbolo habría que aplicar un tipo de cambio, cálculo que podríamos introducir en el "getter". Pero no te preocupes por eso, el ejercicio solo pretende probar el soporte de internacionalización de Spring. Recuerda que el formateo de Spring solo funciona con los campos de formulario, así que para probarlo tendrás que editar o dar de alta un coche.

#### Posibles problemas con la codificación de caracteres

Cuidado con Google Chrome, algunas versiones pueden dar problemas con el símbolo de la moneda. Por otro lado fíjate en que la aplicación usa un filtro para procesar correctamente los caracteres UTF8. Si no lo usáramos se produciría un error al intentar leer caracteres como el del euro. Este filtro está ya implementado en Spring y se llama characterEncodingFilter, puedes ver su definición en el web.xml. No es necesario que modifiques nada.

Ejercicios de validación e internacionalización