Sesión 9. JSF 2.2 (Ajax + Validación + Templates + Snippets)

Índice

Introducción

En esta práctica partimos de la versión de Gestioneitorv7_1 que desarrollamos en la sesión anterior en la que implantamos internacionalización e inyección de dependencia a la aplicación base de gestión de alumnos.

Objetivo

En la presente versión incorporaremos las siguientes mejoras:

- Ajaxificación de los formularios en los que sea oportuno.
- Validación de campos.
- Uso de plantillas para flexibilizar y reducir el trabajo de edición de vistas.
- Incorporación de snippets al uso de plantillas.

Ajax

1. Crear un proyecto nuevo JSF vacío en eclipse seleccionando JSF 2.2 como entorno de ejecución y con nombre tew-gestioneitorv7_1.
2. Descomprimir el zip suministrado sesion9.zip y copiar todas las carpetas a la raíz del proyecto JSF creado en work\tew-gestioneitorv7_1.
3. Para incorporar Ajax en nuestro proyecto tenemos que buscar aquellos formularios que no presenten reglas de transición a otros formularios distintos y que por lo tanto justifiquen el empleo de Ajax. En nuestra aplicación básicamente tenemos sólo un caso que cumple esta condición: en el formulario de listado.xhtml. Sustituye el código del enlace de Baja por el siguiente código:

```
<h:commandLink action="#{controller.baja(valumno)}" type="submit"
value="#{msgs.enlaceBaja}" immediate="true">
    <f:ajax execute="@this" render="@form"/>
</h:commandLink>
```

   Vemos que hemos sustituido el método baja por una versión parametrizada. Además los componentes que se "traen" del servidor vía Ajax son todos los elementos de la tabla (render="@form") y los elementos que se "llevan" al servidor, son sólo el alumno que se pasa como parámetro a controller.baja(valumno), por ello el uso de execute="@this".
   Fíjate en el código HTML generado por JSF en la página listado.xhtml. Analiza en enlace de Baja y busca la función javascript manejadora del evento onclick (mojarra.ab) en la librería JS importada por JSF.
   NOTA MUY IMPORTANTE: Se debe emplear sólo un formulario por vista. En caso de que se desee usar más de un formulario ajaxificado, se debe indicar el id del resto de formularios en la propiedad render de todos los enlaces/botones ajaxificados. Aunque esto lo utilizaremos en el apartado de snippets.
4. Por otro lado, elementos que son ajaxificables habitualmente son las etiquetas internacionalizadas. En este caso, los enlaces de cambio de idioma están en todas las vistas. Si queremos ajaxificar el cambio de todas las etiquetas, sólo tenemos que incluir la etiqueta f:ajax en los `commandLink` de cambio de idioma. Hay dos cuestiones que debes resolver:
   a. ¿Debes usar execute, render o ambos?
   b. ¿A qué elementos de cada vista deseas que afecte la ajaxificación?

Con esta decisión, ajaxifica todos los enlaces de cambio de idioma.

Validación

En este problema vamos a emplear sólo validación explícita automática. Empezaremos por la vista altaForm.xhtml; incorpora los siguientes campos validados:

```
<h:form>

<h:panelGrid columns="2" styleClass="formTable">
  #{msgs.nombre}
  <h:panelGroup>
  <h:inputText value="#{alumno.nombre}"
               required="true"
               requiredMessage="#{msgs.formAlta_nombre_requiredMessage}"
               validatorMessage="#{msgs.formAlta_nombre_validatorMessage}"
               id="nombre">
    <f:validateLength maximum="20"/>
  </h:inputText>
  <h:message for="nombre"/>
  </h:panelGroup>

  #{msgs.apellidos}
  <h:panelGroup>
  <h:inputText value="#{alumno.apellidos}"
               required="true"
               requiredMessage="#{msgs.formAlta_apellidos_requiredMessage}"

validatorMessage="#{msgs.formAlta_apellidos_validatorMessage}"
               id="apellidos">
    <f:validateLength  maximum="40"/>
  </h:inputText>
  <h:message for="apellidos"/>
  </h:panelGroup>

  #{msgs.userid}
  <h:panelGroup>
  <h:inputText value="#{alumno.iduser}"
               required="true"
               requiredMessage="#{msgs.formAlta_iduser_requiredMessage}"
               validatorMessage="#{msgs.formAlta_iduser_validatorMessage}"
               id="iduser">
    <f:validateLength minimum="5" maximum="8"/>
  </h:inputText>
  <h:message for="iduser"/>
  </h:panelGroup>

  #{msgs.correo}
  <h:panelGroup>
  <h:inputText value="#{alumno.email}"
               required="true"
               requiredMessage="#{msgs.formAlta_correo_requiredMessage}"
               validatorMessage="#{msgs.formAlta_correo_validatorMessage}"
               id="correo">
```

```
    <f:validateRegex pattern=".+@.+\.[a-z]+"/>
  </h:inputText>
  <h:message for="correo"/>
  </h:panelGroup>
  </h:panelGrid>
   <h:commandButton value="#{msgs.botonSalvar}"
                    action="#{controller.salva}">
     <!-- Ojo que aquí render tiene que incluir todo el formulario
     ya que los mensajes de error pueden ser renderizados también por Ajax
de form implícita -->
     <f:ajax execute="@form" render="@form"/>
  </h:commandButton>
 …..
</h:form>
```

Al añadir las validaciones de campos, estamos incluyendo nuevos elementos susceptibles de ser renderizados, es decir los elementos h:message. Por ello tiene sentido la ajaxificación de este formulario. Téngase en cuenta que en este formulario es necesario indicar los atributos execute y render:

- Execute lo pondremos a "@form" para ajaxificar el envío de los campos input "nombre", "apellidos", … y que sean validados en el servidor.
- Render debe estar puesto a "@form" ya que los mensajes de error también deben ser ajaxificados para que sean cargados. Prueba a suprimir render="@form" y ver que los mensajes no se cargan.

5. Procede de igual forma con el formulario editForm.xhtml.

Templates

6. Los elementos necesarios para incluir templates en este problema implican cambios sustanciales. Hemos incluido un template general para todas las páginas (copiar de material/templates/template-general.xhml a WebContent/templates, que contiene el siguiente código:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets">
<!-- Si que si quisiera tomar el idioma del navegador -->
<!-- <f:view locale="#{facesContext.externalContext.requestLocale}">  -->
<!-- Para tomar el idioma del bean de configuración beanSettings  -->
<f:view local="#{settings.locale}">
<h:head>
<title>#{msgs.tituloHead}</title>
</h:head>
<h:body>
<center>
<h1>#{msgs.tituloAplicacion1}</h1>
<br/>
<br/>
<h2>#{msgs.tituloAplicacion2}</h2>
</center>
<br/>
<br/>
```

```
<table border="5" align="center">
<tr>
 <th class="title">
   <ui:insert name="titulo">#{msgs.tituloDefectoCabecera}</ui:insert>
</th>
</tr></table>

<ui:insert name="cuerpo">#{msgs.tituloDefectoCuerpo}</ui:insert>

<ui:insert name="pie">#{msgs.tituloPie}</ui:insert>

</h:body>
</f:view>
</html>
```

7. Usaremos este template en todas las vistas de nuestra aplicación. Copiar todos los archivos vista de material/vistas-template/*.xhtml a WebContent (mismas vistas que ya teníamos y que incluyen el uso del template).
8. Veamos a modo de ejemplo la vista del listado de alumnos (listado.xhtml):

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    template="/templates/template-general.xhtml">
<ui:define name="titulo">
    #{msgs.tituloOperacionListado}
</ui:define>

<ui:define name="cuerpo">
<h:form>
<h:dataTable var="valumno"
             value="#{controller.alumnos}"
             border="1">
  <h:column><f:facet
name="header">#{msgs.tablaNombre}</f:facet>#{valumno.nombre}</h:column>
  <h:column><f:facet
name="header">#{msgs.tablaApellidos}</f:facet>#{valumno.apellidos}</h:column>
  <h:column><f:facet
name="header">#{msgs.tablaIdUser}</f:facet>#{valumno.iduser}</h:column>
  <h:column><f:facet
name="header">#{msgs.tablaCorreo}</f:facet>#{valumno.email}</h:column>
  <h:column><f:facet name="header">#{msgs.tablaBaja}</f:facet>
  <h:commandLink action="#{controller.baja(valumno)}" type="submit"
value="#{msgs.enlaceBaja}" immediate="true">
     <f:ajax execute="@this" render="@form"/>
  </h:commandLink>
  </h:column>
  <h:column><f:facet name="header">#{msgs.tablaEditar}</f:facet>
  <h:commandLink action="editar" type="submit" value="#{msgs.enlaceEditar}"
actionListener="#{alumno.setAlumno(valumno)}" immediate="true">
  </h:commandLink>
  </h:column>
</h:dataTable>
<ul>
   <li>
```

```
   <h:commandLink value="#{msgs.enlaceIdiomaES}"
actionListener="#{settings.setSpanish}">
     <f:ajax render="@form"/>
  </h:commandLink>
   </li>
   <li>
   <h:commandLink value="#{msgs.enlaceIdiomaEN}"
actionListener="#{settings.setEnglish}">
     <f:ajax render="@form"/>
   </h:commandLink>
   </li>
   <li><h:commandLink action="casa" value="#{msgs.enlaceCasa}"/> </li>
</ul>
</h:form>
</ui:define>
</ui:composition>
```

Se han definido dos secciones ui:define que sobreescriben las secciones ui:insert correspondientes del template templates-general.xhtml: titulo y cuerpo. De momento el pie no lo hemos incluido ya que implicaría usar dos formularios y eso será lo que hagamos en el siguiente apartado.

Snippets

Dado que el ejemplo que nos ocupa es muy sencillo, el uso de snippets será un poco forzado. No obstante, vamos a hacer una propuesta respetando el template definido en el punto anterior.
En concreto hemos creado tres snippets que tienes en material/snippets, y que debes copiar al directorio WebContent/snippets:

a. form-alumno.xhtml. Contiene un PanelGrid con los campos del alumno. Este snippet está pensando para ser usando en las vistas altaForm.xhtml y editForm.xhtml con el ui:insert body del template.
b. pie-idiomas.xhtml. Contiene un formulario con los enlaces de cambio de idiomas. Este snippet se usará en la vista en index.xhtml, como parte contenido del ui:insert pie.
c. pie-idiomas-casa.xhtml. Contiene el formulario de cambio de idiomas además del enlace a casa. Este snippet se usará en todas las vistas, excepto en index.xhtml, como parte del contenido del ui:insert pie.

9. Copia las vistas de vistas-snippets/* a WebContent y analiza lo que ha cambiado en los atributos render.
   En concreto lo que ha cambiado es que ahora cada vista posee al menos dos formularios:
   - Uno formulario correspondiente a la sección "cuerpo" del template. A este primer formulario le hemos puesto el id="form-principal".
   - Y otro correspondiente a la sección "pie" al que hemos denominado id="form-pie".

   Analicemos a continuación la vista que te suministramos para el alta de alumnos "altaForm.xhtml":

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
```

```
    xmlns:ui="http://java.sun.com/jsf/facelets"
    template="/templates/template-general.xhtml">
<ui:define name="titulo">
    #{msgs.tituloOperacionAlta}
</ui:define>

<ui:define name="cuerpo">
<h:form id="form-principal">
<ui:include src="/snippets/form-alumno.xhtml"/>

     <h:commandButton value="#{msgs.botonSalvar}"
                      action="#{controller.salva}">
     <!-- Ojo que aquí render tiene que incluir todo el formulario
     ya que los mensajes de error pueden ser renderizados también por
Ajax de form implícita -->
     <f:ajax execute="@form" render="@form :form-pie"/>
  </h:commandButton>
</h:form>
</ui:define>

<ui:define name="pie">
   <ui:include src="/snippets/pie-idiomas-casa.xhtml"/>
</ui:define>

</ui:composition>
```

Debido a un bug en la etiqueta f:Ajax en la última versión de JSF (2.2) nos vemos obligados a referenciar explícitamente en el atributo render todos los formularios incluídos en la misma vista. Si te fijas en los snippets que te hemos suministrado, en sus etiquetas Ajax siempre se referencia el formulario ":form-principal". El carácter ':' se debe usar para hacer referencia a elementos externos al formulario en que estamos.

Revisa el código de snippet pie-idiomas-casa.xhtml:

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core">
  <h:form id="form-pie">
   #{msgs.textoIdioma}
    <ul>
       <li>
       <h:commandLink value="#{msgs.enlaceIdiomaES}"
actionListener="#{settings.setSpanish}">
         <f:ajax render="@form :form-principal"/>
       </h:commandLink>
       </li>
       <li>
       <h:commandLink value="#{msgs.enlaceIdiomaEN}"
actionListener="#{settings.setEnglish}">
         <f:ajax render="@form :form-principal"/>
      </h:commandLink>
       </li>
       <li>
```

```
        <h:commandLink value="#{msgs.enlaceCasa}" action="casa">
          <f:ajax render="@form :form-principal"/>
        </h:commandLink>
        </li>
    </ul>
  </h:form>
</ui:composition>
```

Todos las etiquetas f:Ajax llevan como render = "`@form :form-principal", para evitar`

10. Falta por adaptar la vista listado.xhtml y editForm.xhtml.
11. Prueba a quitar alguna referencia a un form externo en los snippets o en las vistas y cuéntame que pasa.