Carrera de Java Programmer SE8

...

Módulo 3 : Desarrollando Aplicaciones Web con JSF

Contenido

- Validación de datos:
  - Custom Validator
- Conversión de datos:
  - Custom Converter
- Uso de tablas de datos (DataTable)
- Manejo de eventos
- Uso de AJAX
- Componentes Compuestos:
  - Etiquetas
- Soporte de HTML 5
- Seguridad en Aplicaciones JSF:
  - Autenticación
  - Autorización
- Manejo de Sesiones:
  - HttpSession
- PrimeFaces

Validación de Datos

Tag	Descripción	Ejemplo
f:validateLength	Valida la longitud de una cadena.	<f:validateLength minimum="5" maximum="8" />
f:validateLongRange	Valida el rango de valor numérico.	<f:validateLongRange minimum="5" maximum="200" />
f:validateDoubleRange	Valida el rango de un valor flotante.	<f:validateDoubleRange minimum="1000.50" maximum="10000.50" />
f:validateRegex	Valida el componente JSF con una expresión regular dada.	<f:validateRegex pattern="PATRON" />
Custom Validator	Validadores personalizados.	<f:validator validatorId=”NOMBRE” />

Validación de Datos: Custom Validator

Pasos:

1. Implementar la interfaz *javax.faces.validator.Validator*
2. Implementar el método *validate()* de la interfaz.
3. Usa el decorador **@FacesValidator** sobre el nombre de la clase, para asignar un Id único al validador personalizado.
4. Implementa tu logica.

```
<h:outputLabel value="Sitio web: " />
<h:inputText id="txtURL" value="#{ persona.web }" label="SitioWeb" >
    <f:validator validatorId="main.UrlValidator" />
</h:inputText>
<h:message for="txtURL" style="color:red" />
```

Ejemplo

Conversión de Datos

Tag	Descripción	Ejemplo
f:convertNumber	Convierte una cadena en un número en un formato dado.	<f:convertNumber minFractionDigits="2" />
f:convertDateTime	Convierte una cadena en una fecha en un formato dado.	<f:convertDateTime pattern="dd-mm-yyyy" />
Custom Convertor	Convertidor personalizado.	<f:converter converterId="NOMBRE" />

Conversión de Datos: Custom Converter

Pasos:

1. Implementar la interfaz *javax.faces.convert.Convert*
2. Implementar los métodos *getAsObject()* y *getAsString()* de la interfaz.
3. Usa el decorador **@FacesConvertor** sobre el nombre de la clase, para asignar un Id único al convertidor personalizado.
4. Implementa tu logica.

```
<!-- Custom Converter -->
<h2>Convertidor Personalizado</h2>
<h:form>
    <h:inputText id="txtUrl" value="#{ persona.url }" label="URL">
      <f:converter converterId="main.UrlConverter" />
   </h:inputText>
   <h:commandButton value="submit" action="resultado"/>
   <h:message for="txtUrl" style="color:red" />
</h:form>
```

Ejemplo

Uso de tablas de datos (DataTable)

Características:

- h:dataTable
  - value
  - rendered
  - var
  - styleClass
  - headerClass
  - rowClasses
- h:column
- f:facet

```
<h:dataTable
    value="#{ eb.lstEmpleados }"
    rendered="#{ not empty eb.lstEmpleados }"
    var="empleado"
    styleClass="employeeTable"
    headerClass="employeeTableHeader"
    rowClasses="employeeTableOddRow, employeeTableEvenRow">
    <h:column>
        <f:facet name="header">Nombre</f:facet>
        #{ empleado.nombre }
    </h:column>
    <h:column>
        <f:facet name="header">Apellido</f:facet>
        #{ empleado.apellido }
    </h:column>
    <h:column>
        <f:facet name="header">Género</f:facet>
        #{ empleado.genero }
    </h:column>
    <h:column>
        <f:facet name="header">Salario</f:facet>
        #{ empleado.salario }
    </h:column>
</h:dataTable>
```

Ejemplo

Manejo de Eventos

Evento	Descripción	Implementación
valueChangeListener	Los eventos de cambio de valor se activan cuando el usuario realiza cambios en los componentes de entrada.	Method Binding Clase: ValueChangeListener Implementar: processValueChange
actionListener	Los eventos de acción se activan cuando el usuario hace clic en un botón o componente de enlace.	Method Binding Clase: ActionListener Implementar: processAction
Application Events	Eventos que se activan durante el ciclo de vida JSF.	PostConstructApplicationEvent, PreDestroyApplicationEvent, PreRenderViewEvent

Ejemplo

Uso de AJAX

Atributo	Descripción
disabled	Si es verdadero, el comportamiento de AJAX se aplicará a cualquier componente principal o secundario. Si es falso, el comportamiento de Ajax se desactivará.
event	El evento que invocará las solicitudes de AJAX, por ejemplo, "hacer clic", "cambiar", "desenfocar", "presionar tecla", etc.
execute	Una lista de identificadores separados por espacios para los componentes que deberían incluirse en la solicitud de Ajax.
immediate	Si los eventos de comportamiento "verdaderos" generados a partir de este comportamiento se transmiten durante la fase Aplicar valores de solicitud. De lo contrario, los eventos se transmitirán durante la fase Invocar aplicaciones.

Uso de AJAX

Atributo	Descripción
Listener	Una expresión EL para un método en un bean de respaldo que se llamará durante la solicitud de Ajax.
Onerror	El nombre de una función de devolución de llamada de JavaScript que se invocará si se produce un error durante la solicitud de Ajax.
Onevent	El nombre de una función de devolución de llamada de JavaScript que se invocará para manejar eventos de IU.
Render	Una lista de identificadores separados por espacios para los componentes que se actualizarán después de una solicitud de Ajax.

Ejemplo

Componentes Compuestos

Características:

- Son mecanismos muy potentes de extensión en JSF.
- No se recurre a código fuente en Java.
- Permite estandarizar la visualización de tus elementos web.
- Es necesario que tus componentes sean creados dentro de la carpeta ***resources*** de tu proyecto.

Indicaciones:

- Dentro de ***resources*** crear una carpeta donde se guardarán tus componentes.
- Para crear un componente se debe utilizar el siguiente espacio de nombre:

xmlns:composite="http://java.sun.com/jsf/composite"

- Para utilizar tus componentes, debes utilizar el espacio de nombres con la siguiente estructura:

xmlns:<NOMBRE>="http://java.sun.com/jsf/composite/<CARPETA>"

Componentes Compuestos: Etiquetas

Etiquetas	Descripción
composite:interface	Declara valores configurables para ser utilizado en composite: implementación.
composite:attribute	Los valores de configuración se declaran utilizando esta etiqueta.
composite:implementation	Declara el componente JSF. Se puede acceder a los valores configurables definidos en composite:interface utilizando la expresión **#{ cc.attrs.attribute-name }**

Ejemplo

Soporte de HTML 5

Soporte:
JSF soporta una gran cantidad de las etiquetas (tags) y atributos del estándar de HTML versión 5.

Se usa el espacio de nombres: xmlns:<NOMBRE>="http://xmlns.jcp.org/jsf/passthrough"

Documentación:
https://docs.oracle.com/javaee/7/tutorial/jsf-facelets009.htm

Ejemplo

Seguridad en Aplicaciones JSF

Autenticación y Autorización

Seguridad en Aplicaciones - Autenticación

Proceso por el cual un usuario o servicio se identifica para poder acceder a ciertos servicios que ofrece nuestro sistema.

Categorías:

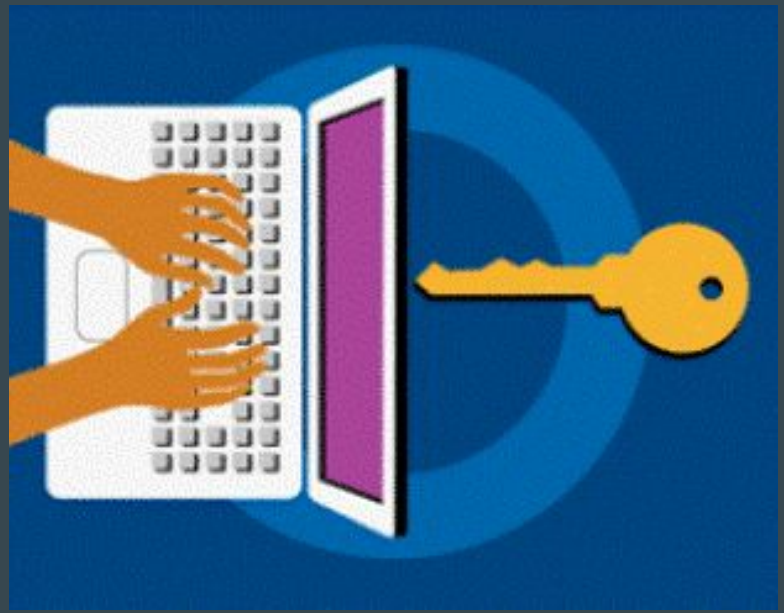
- ¿Qué sabes?: contraseñas, respuestas a preguntas.
- ¿Qué tienes?: TDD, TDC, Tarjeta de Coordenadas.
- ¿Quién eres?: Técnicas biométricas como lectura de retina o huellas dactilares.

Seguridad en Aplicaciones - Autorización

Proceso por el cual se controlan las acciones que tiene un usuario o servicio normalmente ya autenticado, para ello se le conceden o deniegan permisos.

Categorías:

- Declarativa: Gestionados por un administrador independientemente de manera externa al código de la aplicación.
- Programática: Gestionados desde el código fuente de la aplicación.



Manejo de Sesiones

Autentificación: ¿Qué sabes? -> Formulario.

Verificamos los datos ingresados por el usuario:

- Usuario
- Clave

Para persistir la sesión usaremos:

- HttpSession

Autorización: Programática -> Filter

Verificamos el acceso de usuarios autenticados.

- Usar la interfaz **Filter** en ***javax.servlet.Filter***
- Implementar los métodos abstractos:
  - init()
  - doFilter()
  - destroy()
- Usar la anotación **@WebFilter** encima del nombre de la clase:
  - filterName
  - urlPatterns

Manejo de Sesiones - HttpSession

Método	Descripción	Ejemplo
getSession()	Obtener la sesión actual.	FacesContext.getCurrentInstance().getExternalContext().getSession(false);
getAttribute()	Obtener el valor de un atributo.	session.getAttribute("NOMBRE");
setAttribute()	Establecer un atributo.	session.setAttribute("NOMBRE", VALOR);
removeAttribute()	Remover un atributo de la sesión.	session.removeAttribute("NOMBRE");
invalidate()	Invalidar la sesión actual.	session.invalidate();

Ejemplo

PrimeFaces

Características:

- Es una librería de componentes visuales.
- Gran cantidad de componentes visuales listos para usar.
- Soporte nativo para AJAX.
- Soporte para interfaces de usuario para dispositivos móviles.
- Buena documentación.
- Amplia comunidad.
- Es Open Source.
- No requiere configuración.

Como usar:

- Descarga la libreria desde:
  - https://www.primefaces.org/downloads/
  - https://github.com/primefaces/primefaces
- Importar la librería al proyecto.
- Añadir el siguiente namespace:
  - xmlns:p=”http://www.primefaces.org/ui”
- Usar un componente:
  - https://www.primefaces.org/showcase/

Ejemplo