

JSF

librería h: con XHTML

tags

- <h:head>
- <h:body>
- <h:form>

- “action” es la propia página
- Método POST
- Componentes

- entrada
- salida

- <h:inputText> value="#{bean.campo}"
- <h:inputSecret> value="#{bean.campo}"
- <h:commandButton> value="Nombre" action="#{bean.funcion()}"
- <h:inputHidden> variable oculta
- <h:inputTextarea>
- <h:selectManyCheckbox>
- <h:selectManyMenu>
- <h:selectManyListbox>
- <h:outputFormat> mensaje de i18n
- <h:outputLabel> etiqueta
- <h:outputLink>
- <h:outputText> string
- <h:dataTable> value="#{bean.collection}" var="object"

atributos

- converter conversor de datos
- converterMessage mensaje de error por conversión
- dir dirección en que se muestra el mensaje (RTL / LTR)
- label campo de texto multilinea
- lang lenguaje
- required obligatoriedad
- requiredMessage mensaje de advertencia por obligatoriedad
- validator método de validación
- validatorMessage mensaje de error por validación
- valueChangeListener trigger por cambios implements ValueChangeListener event.getNewValue()
- redisplay mantener valor tras recarga

- Acceso a beans
 - #{bean.campo} getters y setters
 - Predefinidas
 - #{sessionScope.objeto.campo}
 - #{requestScope.objeto.campo}
 - #{applicationScope.objeto.campo}

managed beans

- @PostConstruct
 - es posible invocar a un método del bean antes de que se cargue la pagina XHTML
 - debe anotarse con @PostConstruct
 - será un procedimiento sin parámetros
- @ManagedBean(name = "nombreBean", eager = true)
- scopes
 - @SessionScoped
 - @RequestScoped
 - @ApplicationScoped
- return acciones
 - "nombre-face"
 - null regresa a la página de origen
- inyección
 - @ManagedProperty(value="#{bean}")
 - basados en CDI
 - @Named
 - @Inject
- implementar Serializable
- sin atributos públicos
- constructor sin parámetros

jsf

- componentes
 - atributos
 - id
 - identifica al componente
 - opcional
 - auto-generado
 - immediate
 - para inputs y commands
 - default = false
 - fuerza eventos, validadores y conversores
 - rendered
 - determina visibilidad
 - condición
 - style & styleClass CSS
 - value & binding
 - vincula valor con objetos del bean
 - getters y setters
- hiperenlaces <h:link>
 - outcome="href"
 - value="Response"
- JS y CSS
 - <h:outputScript>
 - library="js"
 - name="file.js"
 - <h:outputStyleSheet>
 - library="css"
 - name="file.css"
- core
 - gestión de eventos
 - conversión de datos
 - ítems en una lista
 - validaciones
 - ajax
 - i18n

external context

- para acceder a:
 - session
 - request
 - application
- javax.faces.context.ExternalContext
- FacesContext.getCurrentInstance().getExternalContext()
 - .getSession()
 - .getRequest()

core

- conversores
 - clases
 - javax.faces.convert.Converter
 - BigDecimalConverter
 - BigIntegerConverter
 - BooleanConverter
 - ByteConverter
 - CharacterConverter
 - Float, Double, Enum, Integer...
- ActionListener <f:setPropertyActionListener>
 - target="#{requestScope.obj}"
 - value="#{obj}"
- Validadores <f:validateLongRange>

ajax

- <f:ajax>
 - execute="id"
 - @all
 - @form
 - @this
 - @none
 - render="id"
 - @all
 - @form
 - @this
 - @none
 - event="event"
 - click
 - keyup
 - mouseover
 - focus
 - blur
 - listener="#{bean.funcion}" public void f(AjaxBehaviorEvent);
 - onevent="event"
 - responseXML respuesta en XML
 - responseText respuesta en texto
 - responseCode código numérico
 - source elemento DOM
 - status estado de petición
 - type tipo de llamada
 - onerror="#{bean.funcion}" status:
 - emptyResponse sin respuesta del server
 - httpError error HTTP
 - malformedXML respuesta malformada
 - serverError error por respuesta