

Aplicaciones RIA con ZK

 Sesión 8: Integración con XHTML y servlets

Índice

- ZUML y XHTML
- Objetos impilcitos
- La interfaz Execution

Combinar HTML y ZUML

- Utilizar el espacio de nombres native para enviar las etiquetas al navegador tal cual
- Utilizar el componente HTML para generar etiquetas HTML sencillas
- Utilizar el espacio de nombres XHTML y generar árboles de componentes combinados

Espacio de nombres native

```
<n:ul xmlns:n="http://www.zkoss.org/2005/zk/native">
   <n:li>
   <textbox/>
   </n:li>
   <n:li>
   <textbox/>
   </n:li>
</n:ul>
```



```
<l
  <1i>>
  <input id="z a3 2"/>
  <1i>>
  <input id="z_a3_5"/>
```

El componente HTML

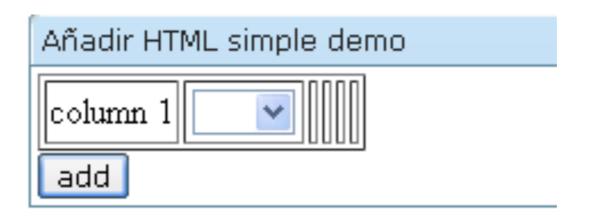
```
<window border="normal" title="Html Demo">
  <html>
  <! [CDATA[
     <h4>Hola, ${parent.title}</h4>
     Este es el contenido del componente html.
   ]]>
  </html>
</window>
```

Puede ser modificado en eventos

```
<zk>
   <html id="h" style="border: 1px solid blue;background: yellow">
   <! [CDATA[
      <l
         Contenido nativo del browser
   ]]>
   </html>
   <button label="change"</pre>
       onClick='h.setContent("Hola, actualizamos")' />
</zk>
```

El espacio de nombres XHTML

- Permite combinar componentes XHTML y componentes ZUL
- Podemos modificar dinámicamente los componentes XHTML



Código

```
<?page title="Añadir HTML" contentType="text/html;charset=UTF-8"?>
\langle zk \rangle
<window id="A" title="Añadir HTML demo" border="normal" xmlns:h="http://</pre>
www.w3.org/1999/xhtml">
    <h:table border="1">
        <h:tr id="row1">
            <h:td>column 1</h:td>
            <h:td>
                 <listbox id="list" mold="select">
                      <listitem label="AA"/>
                      <listitem label="BB"/>
                  </listbox>
              </h:td>
         </h:tr>
     </h:table>
     <button label="add" onClick="row1.appendChild(new org.zkoss.zhtml.Td</pre>
())"/>
</window>
</zk>
```

Componente include

```
<window title="include demo" border="normal" width="300px">
    Hola Mundo!
    <include src="/userguide/misc/includedHello.zul"/>
        <include src="/html/frag.html"/>
        </window>
```

 Es posible pasar un parámetro String a la otra página:

```
<include src="mypage?some=something"/>
```

\${param.some}

O de cualquier tipo de objeto

\${requestScope.some}

Cómo obtener los parámetros HTTP

```
<window>
   1:${param.nombre}
   <button id="b1"/>
   <zscript >
      b1.label = Executions.getCurrent().getParameter("nombre");
   </zscript>
</window>
```

http://localhost:8080/getparameter.zul?nombre=pepito

La interfaz Execution

- Proporciona una capa de abstracción alrededor del concepto de petición del cliente (ServletRequest en las aplicaciones web)
- Permite acceder a datos de la petición:
 - Sesión actual: getSession()
 - Atributos de la sesión: getAttributes()
 - Parámetros de la petición: getParameterMap()

Objetos en donde crear atributos-valor

- Podemos utilizar los métodos getAttribute("nombre") y setAttribute("nombre", "valor") en los siguientes objetos
 - page
 - execution
 - desktop
 - session
 - application

La interfaz Executions

- Proporciona utilidades relacionadas con las peticiones
- Una de ellas es la de crear componentes de páginas **ZUL:**

```
Window win = (Window) Executions.createComponents
("edit.zul", null, null);
```

Ejemplo

