

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)

- JSP 2.2
- EL 2.2

3. Pre-Requisitos

Para el correcto funcionamiento del ejemplo propuesto es imprescindible hacer uso de EL 2.0 o superior debido principalmente al uso de funciones. Para ello la manera más fácil es correr nuestra aplicación bajo Tomcat 7.X que automáticamente nos proporciona soporte para JSP 2.2 Y EL 2.2.

4. Ejemplo práctico

Lo primero de todo y para que sea más cómodo seguir el tutorial os dejo los fuentes del ejemplo aquí . Como vereis a continuación el ejemplo propuesto no puede ser más sencillo, sin embargo es suficiente para ver el potencial de JSTL y EL a la hora de resolver ciertas dificultades a nivel de presentación.

Al acceder a la aplicación <http://localhost:8080/jstl-el/> encontraremos una pantalla inicial con dos enlaces. Ambos enlaces nos mostrarán pantallas iguales a diferencia del soporte de JSTL y EL.



[Página - Sin soporte JSTL EL](#)

[Página - Con soporte JSTL EL](#)

Hacemos "click" en el primer enlace y navegaremos a la siguiente pantalla:



☐ false

cuyo código es el siguiente:

Últimos Tutoriales del Autor

Informes dinámicos con DynamicJasper

Liquibase- Incorporación del histórico de cambios en una BBDD existente

Liquibase-Gestión De Cambios En Base De Datos

Accediendo al repositorio de IFS/SMB , V

in Alfresco S X

ravés



as de

I - Ventas -

lad - ita Contable - BARCELONA.

2011-05-14 Comercial - Ventas - TARRAGONA.

2011-04-13 Comercial - Ventas - VALENCIA.

2011-04-04 Comercial - Compras - CANTABRIA.

view plain print ?

```
01. <?xml version="1.0" encoding="UTF-8"?>
02. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml
03. <html xmlns="http://www.w3.org/1999/xhtml" xmlns:ui="http://java.sun.com/js
04.     xmlns:c="http://java.sun.com/jsp/jstl/core" xmlns:h="http://java.sun.c
05.
06. <h:head>
07.     <meta http-equiv="Content-type" content="text/html; charset=utf-
08.     8" />
09.     <title><h:outputText value="#{msg['app_title']}" /></title>
10. </h:head>
11. <h:body>
12.     <div id="header" style="position:absolute;top:50%;left:50%;margin-
13.     top: -100px;margin-left: -100px;">
14.         <h:graphicImage id="logoAutentia" value="/images/logo.png"/>
15.         <br/>
16.         <div id="content" style="position:relative;top:50%;left:50%;margin-
17.         top:20px;margin-left: -75px;">
18.             <h:form id="form">
19.                 <h:selectBooleanCheckbox id="checkBox" value="#
20.                 {tutoManagedBean.checkBoxNoJslt}" valueChangeListener="#
21.                 {tutoManagedBean.checkBoxNoJsltListener}" style="margin-right:20px;">
22.                     <f:ajax event="change" render="@this form:checkBox"/>
23.                 </h:selectBooleanCheckbox>
24.                 <h:outputText id="valueCheck" value="#
25.                 {tutoManagedBean.checkBoxNoJslt}"/>
26.             </h:form>
27.         </div>
28.     </div>
29. </h:body>
30. </html>
```

Como podeis comprobar esta página no tiene ningun misterio y lo que he pretendido con ella es que podáis observar el distinto comportamiento del componente JSF en ambos navegadores (IE,Firefox) y asi tener una idea exacta del problema que provoca dicho componente. Si os fijáis en la linea 17 es donde definimos el componente JSF con un evento de cambio de valor. Sin embargo si hacéis uso de IE os daréis cuenta de que el cambio de valor se produce cuando el componente pierde el foco, es decir cuando hacemos "click" en el componente y este queda marcado todavía no se ha producido ningun evento y viceversa.

Aquí es donde entra en juego la alianza entre JSTL y EL que nos va a permitir obtener información de la cabecera de la petición. Desde la página de inicio hacemos "click" en el segundo enlace y nos mostrará una pagina exactamente igual que la anterior. Sin embargo si analizamos su código veremos algunas diferencias:

view plain print ?

```
01. <?xml version="1.0" encoding="UTF-8"?>
02. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml
    transitional.dtd">
03. <html xmlns="http://www.w3.org/1999/xhtml" xmlns:ui="http://java.sun.com/jsp
04.       xmlns:c="http://java.sun.com/jsp/jstl/core" xmlns:h="http://java.sun.com
05.
06. <h:head>
07.   <meta http-equiv="Content-type" content="text/html; charset=utf-
08.   8" />
09.   <title><h:outputText value="#{msg['app_title']}" /></title>
10. </h:head>
11.
12. <h:body>
13.   <div id="header" style="position:absolute;top:50%;left:50%;margin-
14.   top: -100px;margin-left: -100px;">
15.     <h:graphicImage id="logoAutentia" value="/images/logo.png"/>
16.     <div id="content" style="position:relative;top:50%;left:50%;margin-
17.     top:20px;margin-left: -75px;">
18.       <h:form id="form">
19.         <h:selectBooleanCheckbox id="checkBox" value="#
20.         {tutoManagedBean.checkBoxJslt}" valueChangeListener="#
21.         {tutoManagedBean.checkBoxJsltListener}" style="margin-right:20px;">
22.         <c:if test="#
23.         {!fn:containsIgnoreCase(request.getHeader('User-Agent'), 'MSIE')}">
24.           <f:ajax event="change" render="@this form:valueCheck" />
25.         </c:if>
26.         <c:if test="#
27.         {fn:containsIgnoreCase(request.getHeader('User-Agent'), 'MSIE')}">
28.           <f:ajax event="click" render="@this form:valueCheck" />
29.         </c:if>
30.         </h:selectBooleanCheckbox>
31.         <h:outputText id="valueCheck" value="#
32.         {tutoManagedBean.checkBoxJslt}" />
33.       </h:form>
34.     </div>
35.   </h:body>
36. </html>
```

Básicamente lo que se pretende con este código es asociar un evento de cambio de valor al componente siempre que el navegador no sea IE. Si el navegador es IE lo que se hace es asociar un evento click para que en ambos navegadores el comportamiento del componente sea el mismo.

En las líneas 3 y 4 se hace referencia a las URL's de los distintos tags que vamos a utilizar a nivel de presentación. La mayoría son los de siempre pero cabe destacar **"http://java.sun.com/jsp/jstl/functions"**. Para completar correctamente el ejemplo es necesario hacer uso de de una función pasandole como parámetro un atributo de la cabecera de la petición.

En la línea 18 vemos como hacer uso de la función `containsIgnoreCase` pasandole como parámetro el atributo `User-Agent` de la cabecera y el valor por el que lo queremos comparar. Esto devolverá true si el `User-Agent` de la cabecera contiene la cadena `MSIE` y false en caso contrario. Además hacemos uso del `if` de JSTL para evaluar el resultado de la función de EL.

Ahora si comprobáis el comportamiento de esta página en ambos navegadores podréis ver que es exactamente el mismo. Eso si cuando hagamos uso de IE la pantalla renderizará un componente con el evento "click" y cuando sea otro navegador se renderizará el componente con el evento "change".

5. Conclusiones.

Seguramente alguno de vosotros estará pensando que no es necesario todo esto para el caso que os he propuesto. Pues bien, nos os falta razón ya que conociendo el comportamiento del componente en IE podríamos haber utilizado el evento "click" y funcionaría correctamente en ambos navegadores pero no quería dejar pasar esta oportunidad de daros a conocer un poquito más acerca de JSTL y EL ya que en más de una ocasión me ha sido verdaderamente útil para resolver ciertas dificultades a nivel de presentación. Por este motivo os recomiendo que le echéis una ojeada al siguiente enlace [Expression Language Specification -Version 2.1](#) para aprender un poco más a fondo como JSTL y EL pueden beneficiaros en vuestros desarrollos.

Espero que les sirva de utilidad.

Un saludo.

Saúl

<mailto:sgdiaz@autentia.com>

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» **Registrate** y accede a esta y otras ventajas «

COMENTARIOS



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Copyright 2003-2011 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

