

¿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)


[Registrarme](#)
[Olvidé mi contraseña](#)
» Estás en: [Inicio](#) » [Tutoriales](#) » [Generación de vistas HTML5 con el soporte de JSF2: pass through](#)[Jose Manuel Sánchez Suárez](#)

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

» [Curso JBoss de Red Hat](#)» [Si eres el responsable o líder técnico, considérate desafortunado. No puedes culpar a nadie por ser gris](#)» [Portales, gestores de contenidos documentales y desarrollos a medida](#)» [Comentando el libro Start-up Nation, La historia del milagro económico de Israel, de Dan Senor & Salu Singer](#)» [Screencasts de programación narrados en Español](#)[Histórico de noticias](#)

Últimos Tutoriales

» [Monta fácilmente tu proyecto con Spring Boot Starter POMs](#)» [\[S.O.L.I.D.\] Dependency inversion principle / Principio de inversión de dependencias](#)Fecha de publicación del tutorial: **2014-10-30**Tutorial visitado 1 veces [Descargar en PDF](#)

Generación de vistas HTML5 con el soporte de JSF2: pass through.

0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno.
- 3. Passthrough attributes: añadir atributos de HTML5 a la renderización de los componentes JSF.
- 4. Passthrough elements: usando HTML5 puro en la vista y JSF como capa de control.
- 5. Referencias.
- 6. Conclusiones.

1. Introducción

El objetivo de este tutorial es muy concreto, ver cómo podemos generar vistas, páginas HTML en versión 5, con el soporte de JSF2.

JSF es el estándar de JEE para la generación de interfaces visuales basadas en componentes reutilizables. Dichos componentes generan, del lado del servidor, todo lo necesario para que en el cliente, en el navegador, tengan el comportamiento y aspecto esperados. Como consecuencia de lo anterior, como desarrolladores, con JSF no tenemos que preocuparnos de cómo se renderizarán estos componentes en el cliente, ni tenemos que trabajar con lenguajes de cliente, véase, HTML y Javascript; los componentes "se pintan" ellos solos.

Es precisamente esta característica una de las reticencias que podemos poner a JSF, puesto que no controlamos cómo se llevará a cabo el renderizado de los componentes, se escapa de nuestro control, o parcialmente de nuestro control. A favor de JSF, las vías a esa falta de control venían siendo:

- siempre podemos usar código HTML en las vistas y ahora no existe ningún problema en combinar html con componentes visuales JSF, y
- siempre podemos usar javascript para añadir comportamiento en cliente a los componentes visuales.

A partir de la versión 2.2 de JSF, a las soluciones anteriores, de andar por casa, se añade el soporte nativo para:

- añadir atributos de HTML5 a la renderización de los componentes JSF y
- usar HTML5 puro en la vista y JSF como capa de control

2. Entorno.

El tutorial está escrito y la instalación realizada usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 15' (2.3 GHz Intel Core i7, 16GB DDR3).
- Sistema Operativo: Mac OS Mavericks 10.9.4
- Eclipse Luna.
- JSF2.2, Mojarra Impl 2.2.8-02

3. Passthrough attributes: añadir atributos de HTML5 a la renderización de los componentes JSF.

El primer soporte para HTML5 es abrir la vía para incorporar a la renderización de cualquier componente JSF atributos del lenguaje cliente a través de lo que han denominado **passthrough**.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE html>
3 <html xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:f="http://xmlns.jcp.org/jsf/core"
5     xmlns:h="http://xmlns.jcp.org/jsf/html"
6     xmlns:ui="http://java.sun.com/jsf/facelets"
7     xmlns:p="http://primefaces.org/ui"
8     xmlns:pt="http://xmlns.jcp.org/jsf/passthrough">
9
10 <ui:composition template="/WEB-INF/templates/generic/layout.xhtml">
11
12 <h:form id="in">
13
14 <h:outputLabel styleClass="ui-label" for="email"
15     value="#{msg.email}" /><br />
16 <p:inputText id="email" size="40"
17     value="#{formView.email}" required="true"
18     pt:type="email" pt:placeholder="Introduzca su email" pt:autofocus="au
19
20 <br />
21
22 <h:outputLabel styleClass="ui-label" for="age"
23     value="#{msg.age}" /><br />
24 <p:inputText id="age" size="40"
25     value="#{formView.age}" required="true"
26     pt:type="number" pt:min="0" pt:max="10" pt:step="2"/>
27
28 <h:outputLabel styleClass="ui-label" for="date"
29     value="#{msg.date}" /><br />
30 <p:inputText id="date" size="40"
31     value="#{formView.date}" required="true"
32     pt:type="date" />
33
34 </h:form>
35
36 </ui:composition>
37 </html>
```



» [S.O.L.I.D.] Interface Segregation Principle / Principio de segregación de interfaz

» [S.O.L.I.D.] Liskov substitution

» [S.O.L.I.D.] Open-Closed Principle / Principio Abierto-Cerrado

Últimos Tutoriales del Autor

» Extendiendo las reglas de SonarQube con Xpath

» Solución de problemas comunes con la integración de maven en Eclipse Luna.

» Integración de SonarQube en Eclipse.

» Monitorización de Apache Tomcat con psi-probe.

» Spring Ldap: gestión de transacciones y configuración de un pool de conexiones

A un componente de entrada de texto básico, aunque en realidad es de Primefaces, estamos incorporando atributos nativos de HTML5, haciendo uso del espacio de nombres **pt**, que viene de passthrough y con ello:

- indicamos que el primer campo es de tipo email, incluyendo un placeholder y un autofocus
- establecemos que el segundo campo es de tipo número estableciendo las características de mínimo, máximo y salto en la selección que decorará el navegador, con un componente nativo del mismo

A continuación una captura de cómo se renderizan en el navegador estos dos componentes.

Email

Edad

Fecha

octubre de 2014 ▾

lun	mar	mié	jue	vie	sáb	dom
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Si en vez de asignar el tipo de componente **number** indicamos **range**

```
1 <p:inputText id="age" size="40"
2     value="#{formView.age}" required="true"
3     pt:type="range" pt:min="0" pt:max="10" pt:step="2"/>
```



La renderización cambiará, en el componente de edad, como sigue:

Edad

Este tipo de atributos no manejados por los componentes JSF, simplemente se incluyen en la fase de renderización al html generado como atributos que son, con lo que los atributos no tienen porque ser necesariamente de HTML5, podrían

ser atributos de cualquier posible framework en cliente.

```
1 <p:inputText id="age" size="40"
2 value="#{formView.age}" required="true"
3 pt:type="range" pt:min="0" pt:max="10" pt:step="2"
4 pt:ng-model="user.age" />
```

El "dejar pasar" la renderización del atributo del componente viene dado por la indefinición y el estado actual de la implementación de la especificación de HTML5 en los navegadores, de tal forma que se dejan pasar tanto los atributos conocidos y los futuros para "que pase cualquiera".

4. Passthrough elements: usando HTML5 puro en la vista y JSF como capa de control.

Si, aún lo anterior, seguimos poniendo reparos a la encapsulación de la renderización a HTML de los componentes JSF, lo cuál es perfectamente entendible, JSF 2.2 nos proporciona una segunda vía, usar directamente HTML5 en la vista pudiendo vincular el estado de los elementos a propiedades en la capa de control.

De este modo, la vista podría estar completamente escrita en HTML5 como sigue:

```
1 <html xmlns="http://www.w3.org/1999/xhtml"
2 xmlns:f="http://xmlns.jcp.org/jsf/core"
3 xmlns:h="http://xmlns.jcp.org/jsf/html"
4 xmlns:jsf="http://xmlns.jcp.org/jsf"
5 xmlns:p="http://primefaces.org/ui">
6 ...
7 <form jsf:id="in">
8
9 <label class="ui-label" jsf:for="email"
10 jsf:value="Email" /><br />
11
12 <input jsf:id="email" size="40"
13 jsf:value="#{formView.name}" data-required="true"
14 type="text" placeholder="Introduzca su email" autofocus="autofocus" /
15
16 <button jsf:action="#{formView.save}"
17 class="ui-icon-disk" >#{msg.action_save}
18 <f:ajax execute="@form" render="@form :out:name" />
19 </button>
20
21
22 <br /><br />
23 <p:messages redisplay="true" autoUpdate="true" />
24
25 </form>
26 ...
27
```

En la generación del árbol de componentes se crea un componente JSF por elemento HTML que incluya un atributo con el espacio de nombres **jsf:**. El tipo de componente que se crea depende del elemento anotado, de tal forma que existe una correlación entre elemento HTML y componente JSF según el tipo del primero. Un elemento `input type="text"` generará un `h:inputText`.

Todos los atributos con el prefijo **jsf:** serán atributos del componente JSF y los que no tienen el espacio de nombres, del elemento HTML.

Es interesante también la posibilidad de imprimir comportamiento Ajax directamente a un elemento HTML, como en el ejemplo estamos haciendo con el elemento `button`.

5. Referencias.

- <http://jsflive.wordpress.com/2013/08/08/jsf22-html5/>
- <http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/HTML5-JSF/html5jsf.html>

6. Conclusiones.

Lo que aparentemente no es más que una manera distinta de gestionar las vistas puede convertirse en una herramienta muy potente a la hora de generar interfaces visuales basadas en maquetas o plantillas; además perdemos la curva de aprendizaje que supone conocer los componentes visuales de JSF puesto que usamos directamente HTML.

Un saludo.

Jose

jmsanchez@autentia.com

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

★★★★★

Por favor, vota +1 o compártelo si te pareció interesante

[+](#) Share | [f](#) [t](#) [in](#) [✉](#) [★](#) [🔒](#) [g+1](#) 0 [g+1](#) 0

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

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



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

PUSH THIS Page Pushers Community Help?

no clicks

0 people brought clicks to this page

++++++++

powered by [karmacracy](#)