

# ¿Qué ofrece Autentia Real Business Solutions S.L?

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

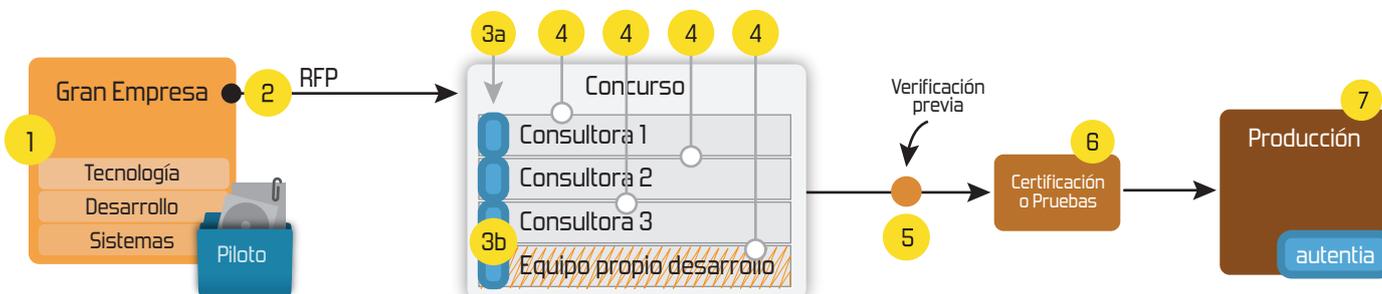
## 1. Desarrollo de componentes y proyectos a medida



## 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)



Estás en: Inicio » Tutoriales » Cómo trabajar con JSF2 y el soporte de inyección de dependencias de Spring



**DESARROLLADO POR:**  
Saúl García Díaz

Consultor tecnológico de desarrollo de proyectos informáticos.

Charla sobre LiquiBase

Charla sobre Alfresco Community Edition

Puedes encontrarme en **Autentia**: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/JEE






Fecha de publicación del tutorial: 2011-08-08

Share |   

Regístrate para votar     

## Cómo trabajar con JSF2 y el soporte de inyección de dependencias de Spring

### 0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno
- 3. Configuración
- 4. A nivel de controlador
- 5. A nivel de interfaz JSF
- 6. Conclusiones

### 1. Introducción

En este tutorial explicaré brevemente como trabajar con JSF2 y el soporte de inyección de dependencias de Spring. El objetivo nos es profundizar en el tema sino fijar la atención en algunos puntos de interes para entender mejor el entorno en el que trabajamos. De esta manera aprovecharemos al máximo las posibilidades que nos ofrecen los frameworks en los que nos apoyamos a la hora de construir nuestras aplicaciones.

### 2. Entorno

El tutorial está escrito usando el siguiente entorno:

- Hardware : Portátil Mac Book Pro 15" (2,6 Ghz Intel Core i7, 4 GB DDR3).
- Sistema Operativo: Mac OS X Snow Leopard 10.6.7
- JSF 2 - Mojarra-2.0.3
- Spring - 3.1.0.M1

### 3. Configuración

En fichero de configuración de JSF faces-config.xml incluimos:

```

view plain print ?
01. ...
02.
03. <el-resolver>org.springframework.web.jsf.el.SpringBeanFacesELResolver</el-resolver>
04.
05. ....
    
```

De esta manera estamos activando el soporte de Spring para la inyección de dependencias de tal forma que JSF sabrá que si no encuentra un bean bajo su contexto debe ir a buscarlo al contexto de Spring. Esto supone una gran ventaja en determinadas situaciones como veremos a continuación.

### 4. A nivel de controlador

Una situación muy habitual que nos encontraremos a la hora de trabajar con JSF2 es la necesidad de realizar inyección de dependencias a nivel de controlador, es decir, desde clases anotadas con **@ManagedBean** y publicadas en el contexto de JSF. Para ello JSF2 nos proporciona la anotación **@ManagedProperty** aunque como nos comentaba nuestro compañero Alex en el tutorial JSF 2 ya está aquí !!! The JSF Return, ahora más sencillo que nunca !!! JSF2 tiene una limitación al respecto ya que solo busca los bean publicados en el directorio clases del proyecto web. Esta limitación nos impediría realizar la inyección de un bean que se encuentre empaquetado en un jar por lo que no podríamos inyectar por ejemplo un bean de negocio.

Otra cosa importante respecto a realizar la inyección con esta anotación es que será necesario incluir el método set del objeto en cuestión que se quiere inyectar.

```

view plain print ?
    
```

Catálogo de servicios Autentia



Últimas Noticias

-  Pirineos on Tour
-  VII Autentia Cycling Day
-  Autentia patrocina la charla sobre Java SE 7 en Madrid
-  Alfresco Day 2011
-  XVII Charla Autentia - Grails - Vídeos y Material

 Histórico de NOTICIAS

Últimos Tutoriales

-  Creando un tema en Liferay 6.0.6
-  Usando CallableStatements para ejecutar procedimientos almacenados
-  Android: Leer correos de Gmail
-  Exportación del contenido de un dataTable de JSF con el soporte de Primefaces
-  Spring schedule annotations

Últimos Tutoriales del Autor

-  Usando CallableStatements para ejecutar procedimientos almacenados
-  JSTL: Leyendo atributos de la cabecera de una petición
-  Informes dinámicos con DynamicJasper
-  Liquibase-Incorporación del histórico de cambios en una BBDD existente
-  Liquibase-Gestión De Cambios En Base De Datos

Síguenos a través de:



```

01. @ManagedBean
02. @ViewScoped
03. public class TutorialView implements Serializable {
04.
05.     ...
06.
07.     @ManagedProperty("#{tutorialNotification}")
08.     private TutorialNotification tutorialNotification;
09.     ...
10.
11.     public void setTutorialNotification(TutorialNotification tutorialNotification) {
12.         this.tutorialNotification = tutorialNotification;
13.     }

```

Es en este punto es donde el soporte de Spring para la inyección de dependencias entra en juego ya que como se explica en el apartado de configuración, JSF sabe que si el la referencia del bean que se necesita inyectar no se encuentra bajo su contexto tendrá que ir a buscarlo al contexto de Spring con lo que queda resuelta la limitación que teníamos sin hacer uso del soporte de Spring y podremos inyectar sin problemas bean que esten empaquetados en un jar fuera del proyecto web.

## 5. A nivel de interfaz JSF

Conociendo un poquito más como funciona el soporte de Spring en cuanto a la inyección de dependencias podemos hacer alguna cosita interesante como por ejemplo hacer uso de un converter y publicarlo en el contexto de Spring con `@Component` en vez de en el contexto de JSF con `@FacesConverter`.

Cuando trabajamos con JSF es muy habitual trabajar con converters a nivel de interfaz de usuario. JSF2 pone a nuestra disposición la anotación `@FacesConverter` para registrar nuestros propios conversores. Pero ¿que ocurre si nuestro conversor necesita inyectar un bean que no se encuentra en el contexto de JSF?. Lo que ocurre es necesitamos hacer uso de nuestro conversor de una manera distinta para poder salvar de nuevo la misma limitación que teníamos en el punto anterior. Lo vemos a continuación:

```

view plain print ?
01. @Component("tutorialConverter")
02. public class TutorialConverter implements Converter{
03.
04.
05.     private TutorialNotification tutorialNotification;
06.
07.     @Autowired
08.     public AlertMessageConverter(TutorialNotification tutorialNotification) {
09.         this.tutorialNotification = tutorialNotification;
10.     }
11.
12.     ....

```

De esta manera estamos publicando nuestro converter en el contexto de Spring y no en el de JSF lo que nos permite inyectar como en este caso por constructor con `@Autowired` referencias a objetos de negocio como servicios. Ahora bien ¿como sabe JSF encontrar nuestro converter sino se encuentra bajo su contexto?. De nuevo entra en juego el soporte de Spring para la inyección de dependencias que hemos configurado y cuando desde la interfaz de usuario hacemos uso del `converter`:

```

view plain print ?
01. ...
02.
03. <h:outputText value="#{tuto.autentia}" converter="tutorialConverter"/>
04.
05. ...

```

JSF busca la referencia para el identificador `tutorialConverter` bajo su contexto y sino lo busca en el contexto de Spring.

## 6. Conclusiones.

Conocer bien como funcionan las distintas piezas que conforman las aplicaciones que desarrollamos hace que tengamos menos limitaciones y más soltura a la hora de llevarlas acabo. Así que espero haber aportado algo de luz sobre este tema y que les pueda servir de utilidad.

Hasta próximos tutoriales.

Un saludo.

Saú I

[sgdiaz@autentia.com](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)

>> **Regístrate** y accede a esta y otras ventajas <<

## Últimas ofertas de empleo

2011-07-06

 Otras Sin catalogar - LUGO.

2011-06-20

 Comercial - Ventas - SEVILLA.

2011-05-24

 Contabilidad - Especialista Contable - BARCELONA.

2011-05-14

 Comercial - Ventas - TARRAGONA.

2011-04-13

 Comercial - Ventas - VALENCIA.

## 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](#)

