

¿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)



E-mail:

Contraseña:

Deseo registrarme
He olvidado mis datos de acceso

Estás en: Inicio » Tutoriales » Spring MVC: acceder a las propiedades de un fichero desde una JSP con Expre...

DESARROLLADO POR:
Miguel Arlandy Rodríguez

Consultor tecnológico de desarrollo de proyectos informáticos.

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

Somos expertos en Java/JEE



Fecha de publicación del tutorial: 2011-07-21

Share |

Regístrate para votar

Spring MVC: acceder a las propiedades de un fichero desde una JSP con Expression Language (EL).

0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno.
- 3. Nuestro fichero de propiedades.
- 4. Bean que almacena las propiedades.
- 5. Configuración del applicationContext.xml.
- 6. Acceso desde la jsp mediante EL (Expression Language).
- 7. Referencias.
- 8. Conclusiones.

1. Introducción

La mayoría de aplicaciones contienen ficheros de configuración (.xml, .properties, etc...) que parametrizan determinados comportamientos de la misma. Los datos de conexión a una base de datos, el máximo número de días para anular un pedido o el importe mínimo de un pedido pueden ser ejemplos de este tipo de propiedades.

En este tutorial vamos a ver cómo acceder a las propiedades de un fichero de configuración con Spring MVC directamente desde una JSP con Lenguaje de Expresiones (EL), haciendo uso de un bean de Spring.

2. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 15' (2.2 Ghz Intel Core I7, 4GB DDR3).
- Sistema Operativo: Mac OS Snow Leopard 10.6.7
- Entorno de desarrollo: Eclipse 3.7 Indigo.
- Spring MVC 3.1.
- Apache Tomcat 6.0.32 con jdk 1.6.

3. Nuestro fichero de propiedades

El primer paso será crear el fichero de propiedades de la aplicación al que llamaremos propiedades.properties.

Imaginemos que nuestra aplicación es una tienda on-line. El fichero contendrá dos propiedades: una para indicar el máximo número de elementos que puede tener un pedido y otra para indicar el importe mínimo para realizar un pedido.

- maximo.elementos.carrito: indica el máximo número de elementos de un pedido.
- importe.minimo: importe mínimo para que se pueda realizar el pedido

El fichero de propiedades quedaría de la siguiente forma:

```
1 maximo.elementos.carrito=15
2 importe.minimo=25.0
```

4. Bean que almacena las propiedades.

Una vez tenemos definidas las propiedades, el siguiente paso será almacenarlas en un bean del que, posteriormente serán consultadas.

Nuestra clase quedaría así:

Catálogo de servicios
Autentia



Últimas Noticias

- VII Autentia Cycling Day
- Autentia patrocina la charla sobre Java SE 7 en Madrid
- Alfresco Day 2011
- XVII Charla Autentia - Grails - Vídeos y Material
- iii 15 millones de descargas de tutoriales !!!

Histórico de NOTICIAS

Últimos Tutoriales

- Framework Scala liftweb
- Trabajando con JAXB y Eclipse
- Configurar Spring Security 3.1 para autenticarse contra un Active Directory
- Migración a ICEfaces 2.0
- Configuración de aplicaciones multitenorio con Maven

Últimos Tutoriales del Autor

- Configurar Spring Security 3.1 para autenticarse contra un Active Directory
- Crear un paginador utilizando JSTL Core
- Implementando nuestro propio formulario de validación con Spring MVC.
- Uso de la Wiki de Github.

Síguenos a través de:



Últimas ofertas de empleo

2011-07-06
 Otras Sin catalogar - LUGO.

Web2PDF

converted by Web2PDFConvert.com

```

01 public class PropiedadesAplicacion {
02     private final Integer maximoElementosCarrito;
03
04     private final Float importeMinimo;
05
06     public PropiedadesAplicacion(Integer maximoElementosCarrito, Float importeMinimo) {
07         Validate.notNull(maximoElementosCarrito);
08         Validate.notNull(importeMinimo);
09         this.maximoElementosCarrito = maximoElementosCarrito;
10         this.importeMinimo = importeMinimo;
11     }
12
13     public Integer getMaximoElementosCarrito() {
14         return maximoElementosCarrito;
15     }
16
17     public Float getImporteMinimo() {
18         return importeMinimo;
19     }
20
21     @Override
22     public int hashCode() {
23         return HashCodeBuilder.reflectionHashCode(this);
24     }
25
26     @Override
27     public boolean equals(Object obj) {
28         return EqualsBuilder.reflectionEquals(this, obj);
29     }
30
31 }
32

```

Como se puede apreciar, el bean consta de dos atributos inmutables que pueden ser consultados mediante sus respectivos métodos get.

5. Configuración del applicationContext.xml.

El siguiente paso será adaptar el fichero de configuración de Spring para que cargue las propiedades del fichero en el bean y lo haga visible, como instancia única, para las jsp's de la aplicación.

En el fichero de configuración debemos informar: dónde está el fichero de propiedades, el bean de propiedades y su registro en el ViewResolver para que sea visible en las vistas (en las jsp's).

```

01 <!-- Ruta de nuestro fichero de propiedades -->
02 <context:property-placeholder location="classpath:propiedades.properties" />
03
04 <!-- Bean en el que se cargan las propiedades que deseemos -->
05 <bean id="propiedadesAplicacion" class="com.autentia.prueba.PropiedadesAplicacion">
06     <constructor-arg name="maximoElementosCarrito" value="${maximo.elementos.carrito}"
07     type="java.lang.Integer"/>
08     <constructor-arg name="importeMinimo" value="${importe.minimo}" type="java.lang.Float"/>
09 </bean>
10
11 <!-- View resolver donde configuramos las vistas y hacemos visible el bean con las propiedades -->
12 <bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">
13     <property name="viewClass" value="org.springframework.web.servlet.view.JstlView" />
14     <property name="prefix" value="/WEB-INF/jsp/" />
15     <property name="suffix" value=".jsp" />
16     <property name="exposedContextBeanNames">
17         <list>
18             <value>propiedadesAplicacion</value>
19         </list>
20     </property>
21 </bean>

```

Obsérvese que para hacer visible el bean a las JSP's de la aplicación, utilizamos la propiedad exposedContextBeanNames del viewResolver.

6. Acceso desde la jsp mediante EL (Expression Language).

En este punto ya estamos en disposición de acceder a las propiedades del bean desde una JSP. La forma de acceder mediante Expression Language sería la siguiente:

```

1 <h3>Accediendo a las propiedades desde una JSP</h3>
2
3 Máximo elementos carrito: ${propiedadesAplicacion.maximoElementosCarrito} <br/>
4 Importe mínimo: ${propiedadesAplicacion.importeMinimo}

```

Y la JSP nos mostraría lo siguiente:

Accediendo a las propiedades desde una JSP


Máximo elementos carrito: 15
Importe mínimo: 25.0

Nótese que el bean PropiedadesAplicacion está en la request, por lo que es posible acceder a él mediante un scriptlet o una expresión con: (PropiedadesAplicacion) request.getAttribute("propiedadesAplicacion"), aunque con Lenguaje de Expresiones el código queda mucho más limpio ;).


7. Referencias.

- Spring MVC: ViewResolver
- Spring Source: InternalResourceViewResolver.html
- Expression Language

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.

8. Conclusiones.

En este tutorial hemos visto cómo acceder a las propiedades de un fichero desde una JSP gracias a un Bean de Spring. Esta solución nos permitirá persistir determinadas propiedades de un fichero (no hay por qué cargar todas las propiedades del fichero en el Bean) para poder ser utilizados en una vista.

Espero que este tutorial os haya sido de ayuda. Un saludo.

Miguel Arlandy

marlandy@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 «

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](#)

XHTML 1.0

CSS

XML RSS

XML ATOM