

¿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

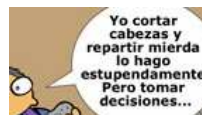
BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)



NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha aquí...

Ver cursos que ofrece Autentia

Descargar comics en PDF y alta resolución



[NUEVO!] 2008-04-20

2008-04-14

2008-04-07

2008-04-01

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la sección [comic](#).

Tutorial desarrollado por



Raúl Expósito Díaz

Consultor tecnológico de desarrollo de proyectos

Catálogo de servicios de Autentia

Descargar (6,2 MB)

Descargar en versión comic (17 MB)

AdictosAlTrabajo.com es el Web de difusión de conocimiento de Autentia.



[Catálogo de cursos](#)

Ingeniero Técnico en Informática de Gestión por la Universidad de Alcalá e Ingeniero en Informática por la Universidad Carlos III de Madrid. [Perfil XING](#)

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/J2EE

Descargar este documento en formato PDF: [springMVCdesdeCero.pdf](#)

Fecha de creación del tutorial: 2008-05-05

Creación de una aplicación web con SpringMVC desde 0

1. Introducción

En este tutorial os vamos a enseñar cómo dar vuestros primeros pasos con Spring MVC y os vamos a dar el código fuente de una aplicación de ejemplo muy básica que os pueda servir como esqueleto.

Tal y como ya ocurrió en el tutorial donde se explica [cómo crear un aplicación con Spring e Hibernate desde 0](#), este tutorial no persigue entrar en conceptos teóricos ni técnicos, ni tampoco pretende desarrollar ninguna funcionalidad de ningún tipo. Simplemente trata de daros un esquema para que sepáis cómo crear una aplicación web usando esta tecnología y podáis dar vuestros primeros pasos.

En el código de este tutorial podéis encontrar un proyecto Eclipse configurado y un pom.xml con todo lo necesario para poder hacerlo funcionar. En el pom.xml se ha configurado jetty para que podáis [arrancar la aplicación directamente desde maven](#). Si queréis acceder al código no tenéis más que pulsar [aquí](#)

2. Entorno

- [Debian](#) GNU/Linux 4.1 (Lenny)
- [JDK 6 Update 1](#)
- [Eclipse 3.3](#) (Europa)
- [Spring 2.5.3](#)

3. ¿Qué necesito saber antes de empezar?

Es conveniente conocer o al menos tener alguna idea del patrón MVC (Modelo-Vista-Controlador), ya que Spring MVC implementa dicho patrón. También es conveniente tener al menos una idea básica de lo que es Spring y de cómo hace la inyección de dependencias.

Si no sabes ninguna de ambas cosas no pasa nada, espero que aún así puedas seguir el tutorial :-)

En el ejemplo vamos a operar sobre los siguientes elementos:

1. El fichero web.xml
2. Ficheros xml para configurar Spring MVC

Catálogo de servicios Autentia (PDF 6,2MB)



En formato comic...



☐ Web

☒ www.adictosaltrabajo.com

Buscar

Últimos tutoriales

2008-05-05
[Creación de una aplicación web con SpringMVC desde 0](#)

2008-05-05
[Cómo integrar Eastwood en nuestras aplicaciones web](#)

2008-04-28
[Cómo lanzar aplicaciones web desde Maven con Jetty](#)

2008-04-28
[Solución al problema de la exportación a HTML de informes JasperReports](#)

2008-04-21
[Proyecto Sakai: Una plataforma de e-learning libre \(II\)](#)

2008-04-21
[Proyecto Sakai: Una plataforma de e-learning libre \(I\)](#)

2008-04-19
[Ampliación de la comparativa de antivirus freeware y opensource](#)

2008-04-19
[Log4j, SMTPAppender: Envío de trazas de log por email](#)

2008-04-17
[Indentación del código fuente](#)

2008-04-16
[Explorar bases de datos HSQLDB](#)

Últimas ofertas de empleo

2008-04-28

3. Controladores muy muy básicos de Spring MVC (lo que en struts son los Action)
4. Páginas jsp

Para simplificar aún más la cosa desde los jsp no pasaremos parámetros a los controladores y desde los controladores no pasaremos parámetros a los jsp.

4. ¿Qué hace la aplicación?

Simplemente muestra dos pantallas y permite navegar entre ellas. Cuando entras en ella, muestra una pantalla sobre **Autentia**:



Más de [60 cursos](#) de formación en nuevas tecnologías

Ver información de [adictosaltrabajo](#)

Si pulsamos sobre 'Ver información de adictosaltrabajo', navegamos hacia otra pantalla:



Más de [550 tutoriales](#) y más de [15 viñetas](#)

Ver información de [Autentia](#)

Si pulsamos sobre 'Ver información de Autentia', navegamos hacia la pantalla anterior, y así en un bucle sin fin.

5. Creación de la aplicación

De nuevo os recuerdo que podeis descargar el código de la aplicación [en este enlace](#).

5.1 web.xml y jsp

Lo primero que vamos a ver es cómo configurar el web.xml para que la aplicación utilice Spring MVC. El código quedaría así:

```
view plain print ?
01. <web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
02. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
03. xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" >
04.
05.     <display-name>Spring MVC desde cero</display-name>
06.
07.     <!-- utilizamos el servlet de spring MVC -->
08.     <servlet>
09.         <servlet-name>autentiaSpringMVC</servlet-name>
10.         <servlet-class>
11.             org.springframework.web.servlet.DispatcherServlet
12.         </servlet-class>
13.         <load-on-startup>1</load-on-startup>
14.     </servlet>
15.
16.     <!-- declaracion de beans utilizados por spring MVC y asociacion entre vistas y controladores -->
17.     <context-param>
18.         <param-name>contextConfigLocation</param-name>
19.         <param-value>/WEB-INF/autentiaSpringMVC-servlet.xml</param-value>
20.     </context-param>
21.
22.     <!-- hacemos que la extension .autentia utilice el servlet de spring -->
23.     <servlet-mapping>
24.         <servlet-name>autentiaSpringMVC</servlet-name>
25.         <url-pattern>*.autentia</url-pattern>
26.     </servlet-mapping>
27.
28.     <welcome-file-list>
29.         <welcome-file>index.jsp</welcome-file>
30.     </welcome-file-list>
31. </web-app>
```

1. Declaramos un servlet llamado autentiaSpringMVC que extiende de una clase de Spring MVC
2. Lo configuraremos en el fichero autentiaSpringMVC-servlet.xml, y es ahí donde vamos a declarar los beans, la navegabilidad, la asociacion entre vistas y controladores, etc.
3. Preparamos el servlet para que, cuando una página acabe en '.autentia', pase por el servlet. Es decir, que vamos a usar Spring MVC en las páginas que acaben por '.autentia'
4. Finalmente indicamos cual es la página de inicio, que es 'index.jsp', cuyo contenido es este y sólo contiene una redirección:

T. Información -
Administrador Sistemas UNIX
/ NT - CIUDAD REAL.

2008-04-23
T. Información - Analista /
Programador - BARCELONA.

2008-04-23
T. Información - Analista /
Programador - BARCELONA.

2008-04-23
T. Información - Analista /
Programador - BARCELONA.

2008-04-23
Otras - Medicina/Farmacia -
SEVILLA.

Anuncios Google

```
view plain print ?
01. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-t
02. <html xmlns="http://www.w3.org/1999/xhtml" >
03. <head>
04. </head>
05. <body onLoad="location.href='autentia.autentia'" >
06. </body>
07. </html>
```

Como podeis ver este jsp no hace nada, simplemente redirecciona a una URL llamada 'autentia.autentia'. La URL termina en '.autentia', de tal modo que el servlet que acabamos de definir va a capturar la invocación y procesarla, más adelante veremos cómo. De momento basta con saber que abrirá el jsp 'autentia.jsp'

```
view plain print ?
01. <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
02.     pageEncoding="ISO-8859-1"%>
03. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd" >
04. <html>
05. <head>
06.     <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" >
07.     <title>Informaci n sobre Autentia</title>
08. </head>
09. <body>
10.     <br/><br/>
11.     
12.     <br/><br/>
13.     <p>M is de <a href="http://www.autentia.com/cursos.php" target="_blank">60 cursos</a> de formac
14.     <p>Ver informaci n de <a href="adictosaltrabajo.autentia">adictosaltrabajo</a></p>
15. </body>
16. </html>
```

En esta pantalla podremos ver la informaci n de Autentia que mencionamos anteriormente. Si pulsamos sobre 'Ver informaci n de adictosaltrabajo' estaremos navegando hacia la URL 'adictosaltrabajo.autentia', que de nuevo ser  procesada por el servlet por acabar en '.autentia'. El servlet en  ltima instancia abrir  el fichero 'adictosaltrabajo.jsp'

```
view plain print ?
01. <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
02.     pageEncoding="ISO-8859-1"%>
03. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd" >
04. <html>
05. <head>
06.     <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" >
07.     <title>Informaci n sobre adictosaltrabajo</title>
08. </head>
09. <body>
10.     <br/><br/>
11.     
12.     <br/><br/>
13.     <p>M is de <a href="http://adictosaltrabajo.com/tutoriales.php" target="_blank">550 tutoriales<
14.     <p>Ver informaci n de <a href="autentia.autentia">Autentia</a></p>
15. </body>
16. </html>
```

5.2 Controladores

Son los hom logos a los Action en Struts. Aunque Spring MVC posee una amplia jerarqu a de controladores yo voy a utilizar en el tutorial el m s b sico, el `AbstractController`. Todos los controladores que extienden de esta clase deben implementar el m todo `handleRequestInternal()`, que se ejecutar  al ser invocado el controlador. Estos controladores no tienen apenas funcionalidad, si quereis usar Spring MVC en vuestros proyectos mirad otros controladores en la documentaci n de Spring ya que seguramente encontreis otros m s  tiles.

```
view plain print ?
01. package com.autentia.tutoriales.springmvc.controladores;
02.
03. import javax.servlet.http.HttpServletRequest;
04. import javax.servlet.http.HttpServletResponse;
05.
06. import org.springframework.web.servlet.ModelAndView;
07. import org.springframework.web.servlet.mvc.AbstractController;
08.
09. public class AutentiaController extends AbstractController {
10.     protected ModelAndView handleRequestInternal(HttpServletRequest arg0, HttpServletResponse arg1) thr
11.         return new ModelAndView("autentia");
12.     }
13. }
14.
```

Este es el c digo de la clase `AutentiaController`. Como veis simplemente devuelve una redirecci n a 'autentia' (ya veremos qu  consigue con eso)

```

view plain print ?
01. package com.autentia.tutoriales.springmvc.controladores;
02.
03. import javax.servlet.http.HttpServletRequest;
04. import javax.servlet.http.HttpServletResponse;
05.
06. import org.springframework.web.servlet.ModelAndView;
07. import org.springframework.web.servlet.mvc.AbstractController;
08.
09. public class AdictosaltrabajoController extends AbstractController {
10.     protected ModelAndView handleRequestInternal(HttpServletRequest arg0, HttpServletResponse arg1) throws
11.         ServletException, IOException {
12.         return new ModelAndView("adictosaltrabajo");
13.     }
14. }

```

Este es el código de AdictosaltrabajoController, que simplemente devuelve una redirección a 'adictosaltrabajo'.

5.3 Configuración de Spring MVC

He aquí cómo juntar todas las piezas: la navegacion, los controladores, los jsp, etc.

```

view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <!DOCTYPE beans PUBLIC "-//SPRING/DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd" >
03.
04. <beans>
05.     <!-- se declaran los controladores en un xml aparte -->
06.     <import resource="resources/controllers.xml" />
07.
08.     <!-- se mapean las url, para dada una direccion, saber qué controlador invocar -->
09.     <bean id="urlMapping" class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping" >
10.         <property name="mappings">
11.             <props>
12.                 <prop key="autentia.autentia">autentiaController</prop>
13.                 <prop key="adictosaltrabajo.autentia">adictosaltrabajoController</prop>
14.             </props>
15.         </property>
16.     </bean>
17.
18.     <!-- genera el nombre del jsp a invocar segun las reglas de navegabilidad de los controladores -->
19.     <bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver" >
20.         <property name="viewClass" value="org.springframework.web.servlet.view.JstlView" />
21.         <property name="prefix" value="/WEB-INF/jsp/" />
22.         <property name="suffix" value=".jsp" />
23.     </bean>
24. </beans>
25.

```

Este es el contenido del fichero resources/controllers.xml

```

view plain print ?
01. <?xml version="1.0" encoding=""?>
02. <!DOCTYPE beans PUBLIC "-//SPRING/DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd" >
03.
04. <beans>
05.     <bean id="autentiaController" class="com.autentia.tutoriales.springmvc.controladores.AutentiaControl
06.     <bean id="adictosaltrabajoController" class="com.autentia.tutoriales.springmvc.controladores.Adictos
07. </beans>
08.

```

1. Se definen los beans que van a representar los controladores.
2. Se mapean las url, para que al acceder a ellas se pase por los controladores que acabamos de definir. Si abres la URL 'autentia.autentia' pasa por el bean 'autentiaController' definido anteriormente, y la URL 'adictosaltrabajo.autentia' por el bean 'adictosaltrabajoController'. Esta asociación se puede hacer de maneras más avanzadas que pueden ser útiles en otras aplicaciones.
3. Se establecen las reglas de navegabilidad, de tal modo que a la salida del controlador se le añade el prefijo '/WEB-INF/jsp/' y el sufijo '.jsp'. De este modo cuando el controlador devuelve 'autentia' se abre el jsp '/WEB-INF/jsp/autentia.jsp'. Esta navegabilidad se puede establecer de otros modos que pueden ser útiles en aplicaciones más complejas.

6. Cómo arrancar la aplicación

Si estás usando maven, basta con ir al directorio donde tengas declarado el pom.xml y ejecutar el siguiente comando:

```
mvn: jetty:run
```

Tras eso, accede a la aplicación desde cualquier navegador a través de esta url:

<http://localhost:8080/autentiaSpringMVC/>

7. Conclusiones

Como podeis ver dar los primeros pasos con Spring MVC es relativamente sencillo ya que, hasta cierto punto, es similar a Struts y otros frameworks MVC. Si ya conoces alguno, Spring MVC te resultará sencillo.

Hay un apartado importante en el que no hemos hecho hincapié, y es que al definir los controladores se pueden inyectar como dependencias cualquier otro tipo de bean y, de este modo, permitirle a los controladores acceder a servicios, utilidades, DAO, etc y de este modo construir una auténtica aplicación web con su acceso a servicios, web services, bases de datos...

Espero que os sea de utilidad.

- Puedes opinar sobre este tutorial haciendo clic aquí.
- Puedes firmar en nuestro libro de visitas haciendo clic aquí.

- Puedes asociarte al grupo [AdictosAlTrabajo](#) en XING haciendo clic aquí.
- Añadir a favoritos [Technorati](#). 



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

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com



Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de subcripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Crear un logger utilizado a través de aspectos con Spring AOP.	En este tutorial os enseñamos cómo implementar un logger utilizado a través de aspectos con Spring AOP.	2008-02-22	989	pdf
Manual básico de Spring WebFlow	En este tutorial Javier Antoniucci nos enseña cómo empezar a trabajar con el framework de desarrollo web Spring webflow.	2007-11-26	2140	pdf
Spring: definición dinámica de Beans	Este tutorial habla sobre la modificación dinámica de los beans del contexto para simplificar la configuración de Spring	2007-05-09	3490	pdf
URLs amigables con Spring MVC	En este tutorial se va a hacer un ejemplo práctico utilizando Spring MVC para la configuración de URLs amigables de nuestra aplicación	2007-04-11	5189	pdf
Cómo realizar pruebas unitarias con Spring y JUnit4 utilizando Gienah	En este tutorial vamos a presentaros Gienah, una tecnología que os permitirá de una forma muy cómoda y sencilla utilizar componentes de Spring en vuestros test unitarios realizados con JUnit 4	2008-02-17	810	pdf
Comparativa entre EJB3 y Spring	En este tutorial os mostramos una comparativa entre EJB3 y Spring esperando que os ayude a decidir qué tecnología utilizar.	2007-10-17	2820	pdf
Spring WebFlow con Validator	En este tutorial se muestra como podemos realizar las validaciones más frecuentes de datos mediante Spring WebFlow.	2007-12-11	1584	pdf
Introducción a Spring Web Flow	Spring Web Flow es un módulo de extensión del framework Spring, que facilita la implementación del flujo de páginas de una aplicación web	2006-01-03	16521	pdf
Creación de una aplicación con Spring e Hibernate desde 0	Este tutorial vamos a explicar paso a paso cómo crear una pequeña aplicación usando Spring e Hibernate con anotaciones partiendo desde 0	2008-02-15	3407	pdf
SpringIDE, plugin de Spring para Eclipse	En adictosaltrabajo os hemos ido presentando diversos plugins para Eclipse. Esta vez le toca el turno a SpringIDE, un plugin que os ayudará a desarrollar aplicaciones que utilicen Spring.	2008-01-19	1876	pdf

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.