

¿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](#) [Subir ficheros al servidor con Spring MVC](#)

Daniel Hernandez del Peso

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática

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

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)PDF
CreatorLast Updated: Aug 23, 2011
License: Free
OS: Windows 7/Vista/2003/XP/2000/NT
Requirements: No special requirements

Fecha de publicación del tutorial: 2012-05-03

Tutorial visitado 1 veces [Descargar en PDF](#)

Subir ficheros al servidor con Spring MVC

Introducción

Los formularios en aplicaciones web son la forma básica de comunicación entre el cliente y el servidor. Una de las cosas que se puede hacer con ellos es subir ficheros para su tratamiento o almacenamiento. Desde el punto de vista de las tecnologías del lado cliente, esto es muy sencillo, puesto que sólo hay que definir un "input" de tipo "file" y un botón de envío del formulario... Las dificultades llegan cuando queremos recoger los datos subidos y tratarlos en la parte servidora.

Estas dificultades se reducen con el uso de frameworks, que nos proporcionan soluciones fáciles y sencillas para este tipo de tareas. A lo largo de los años hemos visto desfilarse por esta página tutoriales para aprender a subir ficheros desde un formulario usando diferentes tecnologías, como el proyecto FileUpload de Jakarta/Apache Commons, Struts, JSF o, incluso, .NET

Spring no va a ser menos y también nos proporciona una manera sencilla de llevar a cabo estas tareas. En este tutorial vamos a ver cómo hacerlo

Entorno del tutorial

El tutorial se ha desarrollado en el siguiente entorno

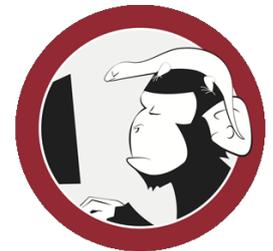
- Mac Book Pro Intel Core i7 2,8GHz
- Sistema operativo Mac OS X 10.6.8 (Snow Leopard)
- Eclipse Indigo (Edición para desarrolladores JEE) con el plugin m2e para integración con Maven

"WebContent/uploadFileSpring/uploadFileSpring.html"

Crear un controlador que pinte un formulario

El primer paso en nuestro tutorial es tener un proyecto de Spring MVC funcionando. Podés hacerlo como querás... Yo lo he creado con el plugin de Maven para eclipse, seleccionando el arquetipo "spring-mvc-webapp"

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

» [Comentando el Libro: Lean StartUp \(el método\)](#)» [VIII Autentia Cycling Day](#)» [¿Por qué una empresa puede dar unos grandes beneficios y tener que despedir gente?](#)» [Técnico en preventa: entre técnico y comercial](#)» [Rotación en la empresa y el teorema de la niña mona](#)[Histórico de noticias](#)

Últimos Tutoriales

» [Cómo instalar Java7 en Mac OS X](#)» [Analiza tu tráfico de red con Wireshark](#)» [Modelando a Pólux en Blender \(3/3\)](#)» [Modelando a Pólux en Blender \(2/3\)](#)» [Primeros pasos con Apache CXF](#)

Últimos Tutoriales del Autor

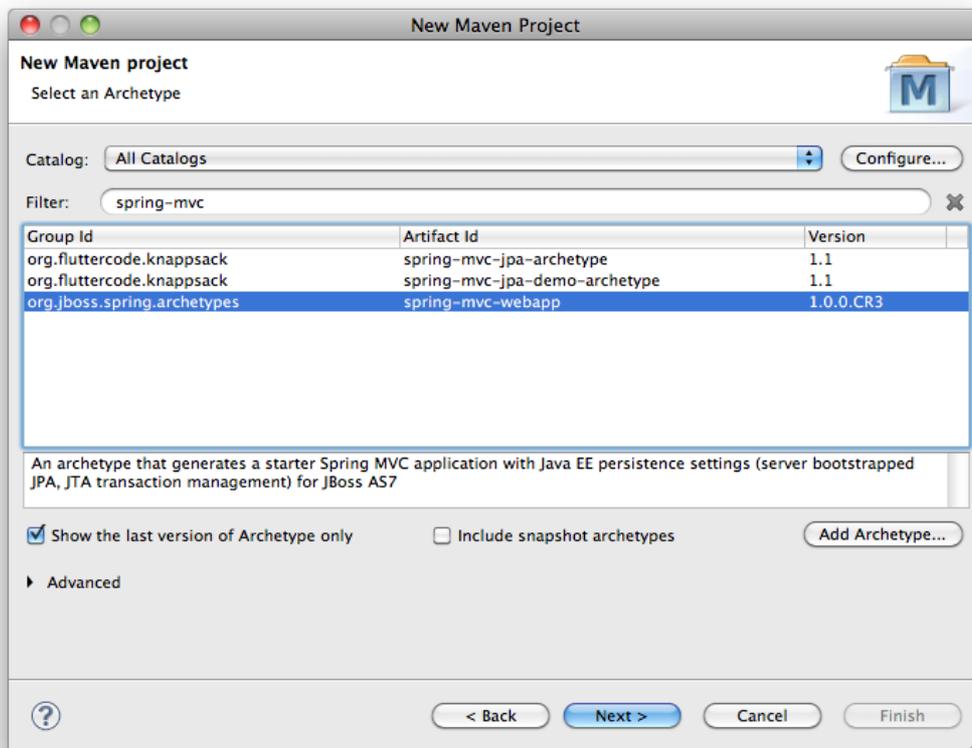
- » Subir una aplicación a BlackBerry App World
- » Transformar aplicaciones Android en aplicaciones BlackBerry
- » Migrar repositorios de Mercurial a Git
- » Creación de cursos con Moodle
- » Administración de sitios Moodle

Categorías del Tutorial

Spring

Últimas ofertas de empleo

- 2011-09-08 [Comercial - Ventas - MADRID.](#)
- 2011-09-03 [Comercial - Ventas - VALENCIA.](#)
- 2011-08-19 [Comercial - Compras - ALICANTE.](#)
- 2011-07-12 [Otras Sin catalogar - MADRID.](#)
- 2011-07-06 [Otras Sin catalogar - LUGO.](#)



Este arquetipo tiene demasiadas funcionalidades para lo que nosotros vamos a necesitar, pero me crea el esqueleto de lo que necesito en cuestión de segundos. Hay que hacer un poco de limpieza para quitar todo lo relativo a JPA, pero el resto os puede valer. Lo importante en este caso es la rapidez :-)

Partiremos de un controlador muy sencillo, que simplemente nos redirigirá a una JSP que pintará el formulario

El controlador

```
1 package com.autentia.tutoriales.file.upload.web;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.RequestMapping;
5 import org.springframework.web.bind.annotation.RequestMethod;
6
7 @Controller
8 @RequestMapping("greet")
9 public class GreetController {
10
11     @RequestMapping(method = RequestMethod.GET)
12     public void getInitialMessage() {
13
14     }
15 }
```

Formulario en JSP (greet.jsp)

```
1 <form method="post" action="greet" enctype="multipart/form-data">
2   <table>
3     <tr>
4       <td>Selecciona fichero: </td>
5       <td><input type="file" name="fichero"></td>
6     </tr>
7     <tr>
8
9     </tr>
10    <tr><td colspan="2" align="center">
11    <input type="submit" value="Subir fichero"></td>
12    </tr>
13  </table>
14 </form>
```

Esto nos da el el siguiente resultado en el navegador:

Selecciona fichero:

Es muy importante el atributo "enctype=multipart/form-data", ya que si no lo ponemos no funcionará el envío de ficheros

Modificar el controlador para gestionar los datos subidos

El primer paso es agregar a nuestra aplicación la capacidad de manejar formularios "multipart". Para ello, instanciamos un "MultipartResolver" como un bean de Spring, en el spring-mvc-context.xml (el fichero donde se van a definir los beans que se emplearán desde los controladores)

```
1 <beans xmlns="...">
```

```

2     xmlns:xsi="..." xmlns:context="..."
3     xmlns:mvc="..."
4     xsi:schemaLocation="...">
5
6     <context:component-scan base-package="com.autentia.tutoriales.file.upload.web" ></context:c
7
8     <mvc:annotation-driven ></mvc:annotation-driven>
9
10    <bean
11        class="org.springframework.web.servlet.view.InternalResourceViewResolver">
12        <property name="prefix" value="/WEB-INF/views/" />
13        <property name="suffix" value=".jsp"></property>
14    </bean>
15
16    <bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultip
17

```

Ahora, nos crearemos una clase (un POJO) que recogerá el fichero del servidor. Esta clase tendrá un único atributo, que se llamará igual que el input de tipo file que hemos puesto en el formulario ("fichero"), con su getter y su setter

```

1     package com.autentia.tutoriales.file.upload.web.formbean;
2
3     import org.springframework.web.multipart.commons.CommonsMultipartFile;
4
5     public class FileFormBean {
6
7         CommonsMultipartFile fichero;
8
9         public CommonsMultipartFile getFichero() {
10            return fichero;
11        }
12
13        public void setFichero(CommonsMultipartFile fichero) {
14            this.fichero = fichero;
15        }
16    }

```

El atributo es de tipo "CommonsMultipartFile", que es la clase que empleará el "resolver" que hemos definido en el xml anteriormente para representar un fichero subido desde el cliente

A continuación, hacemos algunos pequeños cambios tanto en el controlador como en la JSP del formulario

El controlador:

```

1     package com.autentia.tutoriales.file.upload.web;
2
3     import java.io.File;
4     import java.io.FileOutputStream;
5     import java.io.IOException;
6
7     import org.springframework.stereotype.Controller;
8     import org.springframework.web.bind.annotation.ModelAttribute;
9     import org.springframework.web.bind.annotation.RequestMapping;
10    import org.springframework.web.bind.annotation.RequestMethod;
11    import org.springframework.web.multipart.commons.CommonsMultipartFile;
12
13    import com.autentia.tutoriales.file.upload.web.formbean.FileFormBean;
14
15    @Controller
16    @RequestMapping("greet")
17    public class GreetController {
18
19
20
21        @RequestMapping(method = RequestMethod.GET)
22        public @ModelAttribute("fileFormBean") FileFormBean getInitialMessage() {
23            return new FileFormBean();
24        }
25
26        @RequestMapping(method = RequestMethod.POST)
27        public @ModelAttribute("message") String guardaFichero(@ModelAttribute FileFormBean fileFo
28
29            try {
30                grabarFicheroALocal(fileFormBean);
31            } catch (Exception e) {
32                e.printStackTrace();
33                return "No se ha podido grabar el fichero";
34            }
35
36            return "Fichero grabado correctamente";
37        }
38
39        private void grabarFicheroALocal(FileFormBean fileFormBean) throws Exception {
40            CommonsMultipartFile uploaded = fileFormBean.getFichero();
41            File localFile = new File("/Ruta/A/Mi/Fichero/"+uploaded.getOriginalFilename());
42            FileOutputStream os = null;
43
44            try {
45
46                os = new FileOutputStream(localFile);
47                os.write(uploaded.getBytes());
48
49            } finally {
50                if (os != null) {
51                    try {
52                        os.close();
53                    } catch (IOException e) {
54                        e.printStackTrace();
55                    }
56                }
57            }
58        }
59    }

```

El formulario:

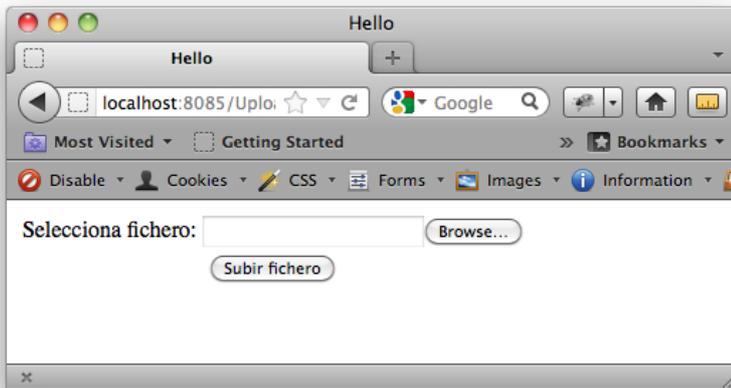
```
1 <%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
2 ...
3 <c:out value="{message}"/>
4 <form:form method="post" action="greet" enctype="multipart/form-data" commandName="fileFormBea
5   <table>
6     <tr>
7       <td>Selecciona fichero: </td>
8       <td><input type="file" name="fichero" /></td>
9     </tr>
10    <tr>
11    </tr>
12    <tr><td colspan="2" align="center">
13    <input type="submit" value="Subir fichero"></td>
14    </tr>
15  </table>
16 </form:form>
```

Además, es necesario agregar al "pom.xml" las dependencias con "commons-fileupload" y "commons-io"

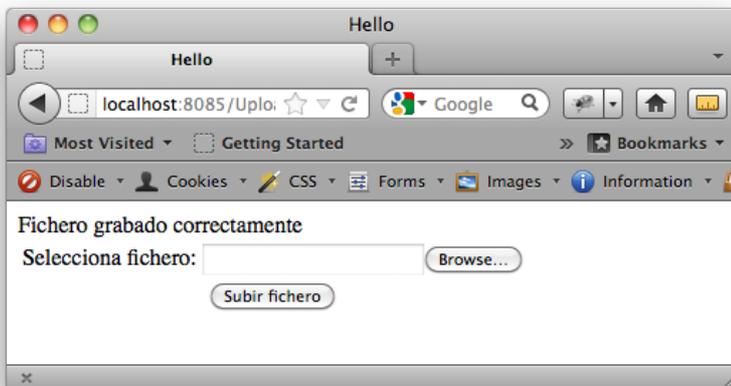
```
1 <dependency>
2   <groupId>commons-fileupload</groupId>
3   <artifactId>commons-fileupload</artifactId>
4   <version>1.2.2</version>
5 </dependency>
6 <dependency>
7   <groupId>commons-io</groupId>
8   <artifactId>commons-io</artifactId>
9   <version>2.0</version>
10 </dependency>
```

Llega el momento de probar...

Tras compilar, empaquetar y redespigar en el servidor, hacemos la prueba. Lanzamos nuestro formulario en el navegador...



... Seleccionamos un fichero y enviamos el formulario, obteniendo un mensaje de éxito



Consultamos el directorio de destino y...

Screen shot 2...2.15.47 PM.png

▼ Preview:



Name Screen shot
2012-04-25 at
12.15.47 PM.png
Kind Portable Network
Graphics image
Size 61 KB on disk

... ¡¡ahí lo tenemos!!

Conclusión

Como hemos visto, con Spring es muy sencillo subir un fichero al servidor... Es importante conocer bien los frameworks que manejamos e incluso los que no manejamos, puesto que en un momento dado nos pueden facilitar de manera decisiva una tarea que de otra manera sería muy costosa

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

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

[Share](#) |

[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](#)

IMPULSA

Impulsores

Comunidad

¿Ayuda?

sin clicks

0 personas han traído clicks a esta página

+ + + + + + + +

powered by [karmacracry](#)