

# Upload Files – Configuración

- 1. Configurar el DispatcherServlet (web.xml) para permitir subir archivos.

Windows

```
<servlet>
  <servlet-name>springmvc</servlet-name>
  <load-on-startup>1</load-on-startup>
  . . .
  <multipart-config>
    <location>c:\tmp</location>
  </multipart-config>
</servlet>
```

Linux

```
<servlet>
  <servlet-name>springmvc</servlet-name>
  <load-on-startup>1</load-on-startup>
  . . .
  <multipart-config>
    <location>/tmp</location>
  </multipart-config>
</servlet>
```

- 2. Declarar un Bean para soporte de MultiPart (upload file). Spring MVC provee la clase StandardServletMultipartResolver (implementación de la interfaz MultipartResolver en la especificación **Servlet 3.0** (Apache Tomcat 7.0.x).

```
<bean id="multipartResolver"
class="org.springframework.web.multipart.support.StandardServletMultipartResolver" />
```

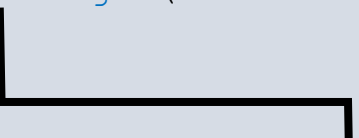
- 3. Agregar el atributo enctype al formulario HTML.

```
<form action="/save" enctype="multipart/form-data">
  <input type="file" name="archivoImagen" />
  <input type="submit">
</form>
```

# Guardar archivo en el disco duro

## Controlador

```
@PostMapping(value = "/save")
public String guardar(... @RequestParam("archivoImagen") MultipartFile multiPart, HttpServletRequest request)
{
    if (!multiPart.isEmpty()) {
        String nombreImagen = guardarImagen(multiPart, request);
    }
    ...
}
```

A black arrow originates from the `guardarImagen(multiPart, request);` line in the `guardar` method and points down to the `guardarImagen` method definition in the block below.

**Método (helper) para guardar el archivo en el disco duro.**

```
private String guardarImagen(MultipartFile multiPart, HttpServletRequest request) {
    // Obtenemos el nombre original del archivo
    String nombreOriginal = multiPart.getOriginalFilename();
    // Obtenemos la ruta ABSOLUTA del directorio images
    // apache-tomcat/webapps/cineapp/resources/images/
    String rutaFinal = request.getServletContext().getRealPath("/resources/images/");
    try {
        // Formamos el nombre del archivo para guardarlo en el disco duro
        File imageFile = new File(rutaFinal + nombreOriginal);
        // Aqui se guarda fisicamente el archivo en el disco duro
        multiPart.transferTo(imageFile);
        return nombreOriginal;
    } catch (IOException e) {
        System.out.println("Error " + e.getMessage());
        return null;
    }
}
```