Spring MVC File Upload Example– Single and Multiple Files

Spring MVC framework provide support for uploading files by integrating Apache Commons FileUpload API. The process to upload files is very easy and requires simple configurations. We will create a simple Spring MVC project in STS.

Most of the part is the boiler-plate code generated by STS tool, we will focus into the changes that are required to utilize Spring file upload integration.

**Step 1:**

### Maven Dependencies for Apache Commons FileUpload

First of all we need to add Apache Commons FileUpload dependencies in our pom.xml file, so that required jar files are part of the web application.

|  |  |
| --- | --- |
| **pom.xml** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <!-- Apache Commons FileUpload -->  <dependency>      <groupId>commons-fileupload</groupId>      <artifactId>commons-fileupload</artifactId>      <version>1.3.1</version>  </dependency>    <!-- Apache Commons IO -->  <dependency>      <groupId>commons-io</groupId>      <artifactId>commons-io</artifactId>      <version>2.4</version>  </dependency> |

### Step 2: File Upload Form Views

We will create two JSP pages to allow single and multiple file uploads.

upload.jsp

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@ page session="false" %>

<html>

<head>

<title>Upload File Request Page</title>

</head>

<body>

    <form method="POST" action="uploadFile" enctype="multipart/form-data">

        File to upload: <input type="file" name="file"><br />

        Name: <input type="text" name="name"><br /> <br />

        <input type="submit" value="Upload"> Press here to upload the file!

    </form>

</body>

</html>

****

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@ page session="false" %>

<html>

<head>

<title>Upload Multiple File Request Page</title>

</head>

<body>

    <form method="POST" action="uploadMultipleFile" enctype="multipart/form-data">

        File1 to upload: <input type="file" name="file"><br />

        Name1: <input type="text" name="name"><br /> <br />

        File2 to upload: <input type="file" name="file"><br />

        Name2: <input type="text" name="name"><br /> <br />

        <input type="submit" value="Upload"> Press here to upload the file!

    </form>

</body>

</html>

**NOTE:** Form **enctype** should be **multipart/form-data**, so that Spring web application knows that the request contains file data that needs to be processed.

Also note that for multiple files, the form field “file” and “name” are same in the input fields, so that the data will be sent in the form of array. We will take the input array and parse the file data and store it in the given file name.

### Spring Configuration for Multipart

To utilize Apache Commons FileUpload for handling multipart requests, all we need to do is configuremultipartResolver bean with class as org.springframework.web.multipart.commons.CommonsMultipartResolver.

****

I am setting maximum upload size limit by providing the maxUploadSize property value formultipartResolver bean. If you will look into the source code of DispatcherServlet class, you will see that a MultipartResolver variable with name multipartResolver is defined and initialized in below method.