

Home &gt; Spring &gt; Spring 3 MVC

## Spring MVC Multiple File Upload Example

By [Viral Patel](#) on November 7, 2012

In this simple tutorial we will see how to implement multiple file upload in a Spring 3 MVC based application.

The requirement is simple. We have a form which displays file input component. User selects a file and upload it. Also its possible to add more file input components using Add button. Once the files are selected and uploaded, the file names are displayed on success page.

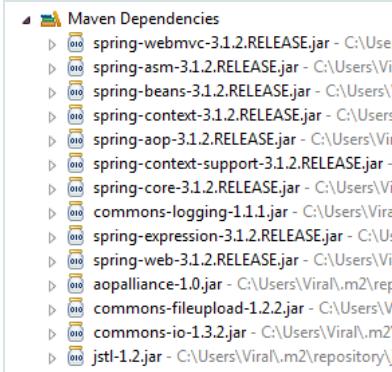
### 1. Maven Dependencies / Required JAR files

If you using Maven in your project for dependency management, you'll need to add dependencies for Apache Common File upload and Apache Common IO libraries. The spring's [CommonsMultipartResolver](#) class internal uses these library to handle uploaded content.

Add following dependencies in your maven based project to add File upload feature.

```
<dependencies>
    <!-- Spring 3 MVC -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
        <version>3.1.2.RELEASE</version>
    </dependency>
    <!-- Apache Commons file upload -->
    <dependency>
        <groupId>commons-fileupload</groupId>
        <artifactId>commons-fileupload</artifactId>
        <version>1.2.2</version>
    </dependency>
    <!-- Apache Commons IO -->
    <dependency>
        <groupId>org.apache.commons</groupId>
        <artifactId>commons-io</artifactId>
        <version>1.3.2</version>
    </dependency>
    <!-- JSTL for c: tag -->
    <dependency>
        <groupId>jstl</groupId>
        <artifactId>jstl</artifactId>
        <version>1.2</version>
    </dependency>
</dependencies>
```

If you have a simple web application, add following JAR files in WEB-INF/lib folder. You can download all these JARs with source code at the end of this tutorial.



### 2. Model – Form Object

Create a Java bean which acts as Model/Form object for our Spring application. This bean contains a [List](#) of [org.springframework.web.multipart.MultipartFile](#) objects. Spring framework provides a useful class [MultipartFile](#) which can be used to fetch the file content of uploaded file. Apart from its content, the [MultipartFile](#) object also gives you other useful information such as filename, file size etc.

*FileUploadForm.java*

```
package net.viralpatel.spring3.form;
```

[Advertise Here](#)

[Subscribe](#)

Get our Articles via Email. Enter your email.

Your E-Mail

Viral Patel

780 followers

[Follow on Twitter @viralpatelnet](#)

[Latest Posts](#)

1. [HTML5 DataList Example](#)
2. [Auditing DML changes in Oracle](#)
3. [Java 8 Lambda Expressions Tutorial with Examples](#)
4. [Java 8 Default Methods Tutorial](#)
5. [Compound Triggers in Oracle 11g - Tutorial with example](#)
6. [How to pass CLOB argument in EXECUTE IMMEDIATE](#)
7. [How to get Eclipse current workspace path](#)
8. [jQuery window height is not correct](#)
9. [45 Useful Oracle Queries](#)
10. [JavaScript Singleton Design Pattern](#)

```
import java.util.List;
import org.springframework.web.multipart.MultipartFile;
public class FileUploadForm {
    private List<MultipartFile> files;
    //Getter and setter methods
}
```

### 3. Controller – Spring Controller

Create a Spring 3 MVC based controller which handles file upload. There are two methods in this controller:

1. `displayForm` – Is used to show input form to user. It simply forwards to the page `file_upload_form.jsp`
2. `save` – Fetches the form using `@ModelAttribute` annotation and get the File content from it. It creates a list of filenames of files being uploaded and pass this list to success page.

*FileUploadController.java*

```
package net.viralpatel.spring3.controller;
import java.util.ArrayList;
import java.util.List;
import net.viralpatel.spring3.form.FileUploadForm;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.multipart.MultipartFile;
@Controller
public class FileUploadController {
    @RequestMapping(value = "/show", method = RequestMethod.GET)
    public String displayForm() {
        return "file_upload_form";
    }
    @RequestMapping(value = "/save", method = RequestMethod.POST)
    public String save(
        @ModelAttribute("uploadForm") FileUploadForm uploadForm,
        Model map) {
        List<MultipartFile> files = uploadForm.getFiles();
        List<String> fileNames = new ArrayList<String>();
        if(null != files && files.size() > 0) {
            for (MultipartFile multipartFile : files) {
                String fileName = multipartFile.getOriginalFilename();
                fileNames.add(fileName);
                //Handle file content - multipartFile.getInputStream()
            }
        }
        map.addAttribute("files", fileNames);
        return "file_upload_success";
    }
}
```

### 4. View – JSP views

Now create the view pages for this application. We will need two JSPs, one to display file upload form and another to show result on successful upload.

The `file_upload_form.jsp` displays a form with file input. Apart from this we have added small jquery snippet onclick of Add button. This will add a new file input component at the end of form. This allows user to upload as many files as they want (subjected to file size limit ofcourse).

Note that we have set `enctype="multipart/form-data"` attribute of our `<form>` tag.

*file\_upload\_form.jsp*

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<head>
    <title>Spring MVC Multiple File Upload</title>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
<script>
$(document).ready(function() {
    //add more file components if Add is clicked
})
```

```
$('#addFile').click(function() {
    var fileIndex = $('#fileTable tr').children().length - 1;
    $('#fileTable').append(
        '<tr><td>' +
        '    <input type="file" name="files['+ fileIndex +']" />' +
        '</td></tr>');
});

</script>
</head>
<body>
<h1>Spring Multiple File Upload example</h1>

<form:form method="post" action="save.html"
    modelAttribute="uploadForm" enctype="multipart/form-data">

    <p>Select files to upload. Press Add button to add more file inputs.</p>

    <input id="addFile" type="button" value="Add File" />
    <table id="fileTable">
        <tr>
            <td><input name="files[0]" type="file" /></td>
        </tr>
        <tr>
            <td><input name="files[1]" type="file" /></td>
        </tr>
    </table>
    <br/><input type="submit" value="Upload" />
</form:form>
</body>
</html>
```

Note that we defined the file input name as **files[0]**, **files[1]** etc. This will map the submitted files to the List object correctly.

I would suggest you to go through this tutorial to understand how Spring maps multiple entries from form to bean: [Multiple Row Form Submit using List of Beans](#)

Second view page is to display filename of uploaded file. It simply loops through filename list and display the names.

*file\_upload\_success.jsp*

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<html>
<head>
    <title>Spring MVC Multiple File Upload</title>
</head>
<body>
    <h1>Spring Multiple File Upload example</h1>
    <p>Following files are uploaded successfully.</p>
    <ol>
        <c:forEach items="${files}" var="file">
            <li>${file}</li>
        </c:forEach>
    </ol>
</body>
</html>
```

## 5. Spring Configuration

In Spring configuration (spring-servlet.xml) we define several important configurations. Note how we defined bean **multipartResolver**. This will make sure Spring handles the file upload correctly using **CommonsMultipartResolver** class.

*spring-servlet.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
        <context:annotation-config />
        <context:component-scan base-package="net.viralpatel.spring3.controller"/>

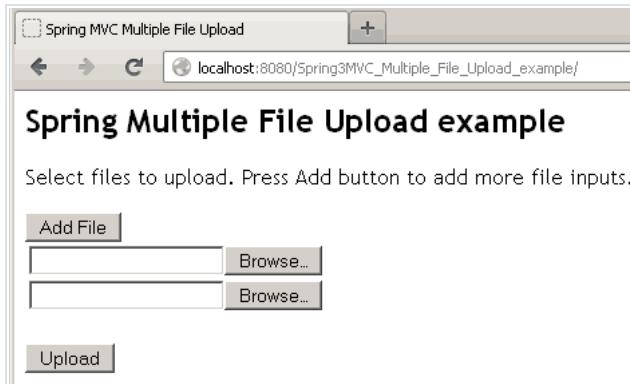
<bean id="multipartResolver"
    class="org.springframework.web.multipart.commons.CommonsMultipartResolver"/>

<bean id="jspViewResolver"
    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="viewClass"
            value="org.springframework.web.servlet.view.JstlView" />
        <property name="prefix" value="/WEB-INF/jsp/" />
        <property name="suffix" value=".jsp" />
    </bean>
</beans>
```

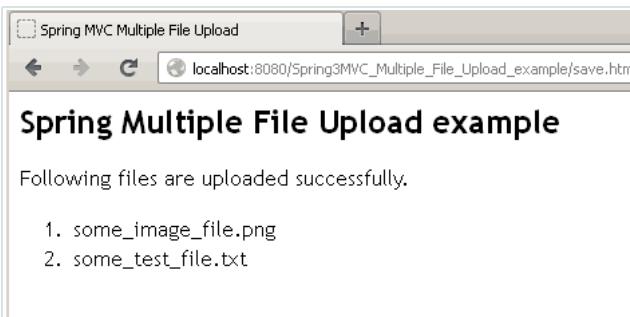
## 6. Output

Execute the project in Eclipse. Open following URL in browser to see file upload form.

URL: [http://localhost:8080/Spring3MVC\\_Multiple\\_File\\_Upload\\_example/show.html](http://localhost:8080/Spring3MVC_Multiple_File_Upload_example/show.html)



Select files through file dialog and press Upload button to upload. Following page will displayed with list of files being uploaded.



We have added a small Javascript snippet for **Add** button. This will add more file upload components to the page. Use this if you want to upload more files.

## Download Source Code

[SpringMVC\\_Multi\\_File\\_Upload\\_example.zip \(3.6 MB\)](#)

## Related Articles

1. [Spring MVC: Multiple Row Form Submit using List of Beans](#)
2. [Spring @RequestHeader Annotation example](#)
3. [Spring 3 MVC: Handling Forms in Spring 3.0 MVC](#)
4. [Spring 3 MVC: Create Hello World application in Spring 3.0 MVC](#)
5. [Spring MVC Flash Attribute tutorial with example](#)
6. [Spring MVC HashMap Form Integration example](#)
7. [Spring 3 MVC Interceptor tutorial with example](#)

Get our Articles via Email. Enter your email address.

Tags: [FILE-UPLOAD](#), [SPRING MVC](#), [SPRING-3-MVC-SERIES](#)

## 32 Comments



**Yogesh**

22 November, 2012, 10:21

Hi Viral! A great tutorial! How do we restrict the file types being uploaded. Use case being i want the user to upload only .pdf file or .xls files.

[Reply](#)

**Greg**

You can use a validator, first register your validator

## Spring MVC Multiple File Upload tutorial. Spring Multiple file upload example

```
@InitBinder("uploadItem")
void initBinder(WebDataBinder binder) {
    binder.setValidator(new UploadValidator ());
}
```

25 January, 2013, 18:41

Then you apply that validator to uploaded file

```
public String uploadPluginAction(@Valid @ModelAttribute("uploadItem") FileUploadBean uploadItem, BindingResult result, ModelMap
        HttpServletRequest request, HttpServletResponse response)
{
    logger.trace("Upload Handler");
    // Check for validation error
    if (result.hasErrors())
    {

        return "/add";
    }
}

import java.io.IOException;

import org.apache.log4j.Logger;
import org.springframework.validation.Errors;
import org.springframework.validation.Validator;
import org.springframework.web.multipart.MultipartFile;

public class UploadValidator implements Validator {
    private static transient Logger logger = Logger.getLogger(UploadValidator.class);
    // Content types the user can upload
    private static final String[] ACCEPTED_CONTENT_TYPES = new String[] {
        "application/pdf",
        "application/doc",
        "application/msword",
        "application/rtf",
        "text/richtext",
        "text/rtf",
        "text/plain",
        "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
        "application/vnd.sun.xml.writer",
        "application/x-soffice",
    };

    private static final String[] ACCEPTED_EXTENSIONS = new String[] {
        "doc",
        "pdf",
        "docx",
        "rtf",
        "txt",
    };

    @Override
    public boolean supports(Class<?> clazz) {
        return FileUploadBean.class.equals(clazz);
    }

    /**
     * Validate our uploaded bean
     */
    @Override
    public void validate(Object obj, Errors errors) {
        FileUploadBean uploadItem = (FileUploadBean) obj;
        MultipartFile file = uploadItem.getFile();
        try {
            if(file == null || file.getBytes().length == 0){
                errors.reject("error.upload.null", "File name can't be empty");
                return;
            }
        } catch (IOException e) {
            logger.error(e.getMessage());
        }

        // Check content type
        boolean acceptableContentType = false;
        String incomingContentType = file.getContentType();
        logger.info("FileName = "+file.getName());
        logger.info("incomingContentType = "+incomingContentType);
        // This related to bug when on Vista and using Firefox content type is 'application/octet-stream'
        // Instead of one of the allowed ones
        if("application/octet-stream".equalsIgnoreCase(incomingContentType)){
            int index = file.getOriginalFilename().lastIndexOf('.');
            if(index > -1 && index < file.getOriginalFilename().length - 1)
                acceptableContentType = true;
        }
    }
}
```



Aks13

22 January, 2013, 1:44

Spring Multiple File Upload example  
 Following files are uploaded successfully.  
 CreateDB.sql  
 Where is place uploaded files ?

[Reply](#)warika

22 January, 2013, 22:09

I want to using ajax(not submit form), read the selected file in input file. You can help me...

[Reply](#)Asier

2 July, 2013, 21:52

Hi,  
 I need to send a similar form but using AJAX. Any solution?  
 Thanks a lot!

[Reply](#)rudi

3 February, 2013, 16:08

hai all  
 can you help me now?  
 For some of my homework (one of many problems)  
 "how to uploading file xls with asp mvc 3 to database?"  
 share with me please, simple sample  
 programming ASP MVC3  
 Database SQL SERVER  
 example databases name\_DB = db\_school, name\_table = tbl\_student, Field = - id, name

[Reply](#)Sid

4 February, 2013, 10:29

Thanks for detailed explanation. Before visiting to this page I didn't knew anything about file upload,  
 Now i can write file upload code easily..  
 Regards  
 sid

[Reply](#)Sudipto Saha

20 February, 2013, 15:40

I was not sure where the files were uploaded...  
 So,I just got your code a little modified with d help of some other example:  
 And here is the full running code:

- 1.Just Create D:\TestUpload directory structure or whatever.
- 2.Copy this code to FileUploadController.java

```
package net.viralpatel.spring3.controller;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import net.viralpatel.spring3.form.FileUploadForm;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.multipart.MultipartFile;

@Controller
public class FileUploadController {
    private String saveDirectory = "D:/Test/Upload/"; //Here I Added

    @RequestMapping(value = "/show", method = RequestMethod.GET)
    public String displayForm() {
        return "file_upload_form";
    }
}
```

## Spring MVC Multiple File Upload tutorial. Spring Multiple File upload example

```

@RequestMapping(value = "/save", method = RequestMethod.POST)
public String save(
        @ModelAttribute("uploadForm") FileUploadForm uploadForm,
        Model map) throws IllegalStateException, IOException {

    List<MultipartFile> files = uploadForm.getFiles();

    List<String> fileNames = new ArrayList<String>();

    if(null != files && files.size() > 0) {
        for (MultipartFile multipartFile : files) {

            String fileName = multipartFile.getOriginalFilename();
            fileNames.add(fileName);
            //Handle file content - multipartFile.getInputStream()
            multipartFile.transferTo(new File(saveDirectory + multipartFile.getOriginalFilename())); //Here I Added
        }
    }

    map.addAttribute("files", fileNames);
    return "file_upload_success";
}
}

```

[Reply](#)


**SATHYA**

30 May, 2013, 16:29

when I copied your updated controller, the file is getting uploaded into the location. But, I'm getting this error. Please clarify.

```

java.io.IOException: Destination file [C:\Temp\Upload] already exists and could not be
deleted
org.springframework.web.multipart.commons.CommonsMultipartFile.transferTo(CommonsMultipartFile.java:130)

```

[Reply](#)


**SATHYA**

30 May, 2013, 17:28

I got the solution for the above exception. I just added a condition like this and added the files in the last step. Please see the code below:

```

if(null != files && files.size() > 0) {
    for (MultipartFile multipartFile : files) {

        String fileName = multipartFile.getOriginalFilename();
        if(!"".equalsIgnoreCase(fileName)){
            //Handle file content - multipartFile.getInputStream()
            multipartFile.transferTo(new File(saveDirectory + fileName)); //Here I Added
            fileNames.add(fileName);
        }
    }
}

```

[Reply](#)


**Suraj**

17 September, 2013, 15:22

This was helpful :)

[Reply](#)


**Dhoni**

4 March, 2013, 22:18

cool machi...yenga tale viral-kku oru ooooooo podu...

[Reply](#)


**prem**

28 March, 2013, 11:52

Hi viral could you please rectify this following exception

```

SEVERE: Context initialization failed
org.springframework.beans.factory.xml.XmlBeanDefinitionStoreException: Line 12 in XML document from ServletContext resource [/WEB-INF/
        at org.springframework.beans.factory.xml.XmlBeanDefinitionReader.doLoadBeanDefinitions(XmlBeanDefinitionReader.java:396)
        at org.springframework.beans.factory.xml.XmlBeanDefinitionReader.loadBeanDefinitions(XmlBeanDefinitionReader.java:334)
SEVERE: WebModule[/SpringFileUpload]StandardWrapper.Throwable
org.springframework.beans.factory.xml.XmlBeanDefinitionStoreException: Line 12 in XML document from ServletContext resource [/WEB-INF/
        at org.springframework.beans.factory.xml.XmlBeanDefinitionReader.doLoadBeanDefinitions(XmlBeanDefinitionReader.java:396)

```

**Spring MVC Multiple File Upload tutorial. Spring Multiple File upload example**

```

at java.lang.Thread.run(Thread.java:619)

SEVERE: Exception while invoking class com.sun.enterprise.web.WebApplication start method
java.lang.Exception: java.lang.IllegalStateException: ContainerBase.addChild: start: org.apache.catalina.LifecycleException: org.apach
at com.sun.enterprise.web.WebApplication.start(WebApplication.java:138)
at org.glassfish.internal.data.EngineRef.start(EngineRef.java:130)

SEVERE: Exception while loading the app
SEVERE: Exception while loading the app : java.lang.IllegalStateException: ContainerBase.addChild: start: org.apache.catalina.Lifecycl

```

[Reply](#)**Vamshi**

18 April, 2013, 0:08

This really helped me. Struggled lot of time to get this done before seeing this tutorial/topic  
Thankyou.

[Reply](#)**sumeet singh**

27 April, 2013, 11:28

very nice tutorial for beginners.

[Reply](#)**praveen**

29 May, 2013, 22:58

Really a great tutorial. I am getting file name as null in controller. here is my controller code.

```

List files = fileUploadForm.getFiles();
List fileNames = new ArrayList();
BufferedReader br;
FileWriter fstream;
BufferedWriter out = null;

if(null != files && files.size() > 0) {
    for (MultipartFile multipartFile : files) {

        String fileName = multipartFile.getOriginalFilename();
        fileNames.add(fileName);
        System.out.println(fileNames.get(0));
        try {

            fstream = new FileWriter("D://input.txt");
            out = new BufferedWriter(fstream);

            InputStream inputStream = multipartFile.getInputStream();

            br = new BufferedReader(new InputStreamReader(inputStream));

            while(br.readLine()!=null){
                System.out.println("line "+br.readLine());
                out.write(br.readLine().toString());
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        finally{
            try {
                out.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

Please help me

[Reply](#)**Dinesh**

24 June, 2013, 18:16

Check if you have added  
in spring-servlet.xml

[Reply](#)

**Dinesh**

24 June, 2013, 18:21

```
<bean id="multipartResolver"
      class="org.springframework.web.multipart.commons.CommonsMultipartResolver" />
```

[Reply](#)**Thasnim hm**

19 June, 2013, 20:27

Hi,  
Thanks for sharing. Nice one .

[Reply](#)**Dinesh**

25 June, 2013, 10:21

Hi, dynamically added file type not working. when I click on add file and upload three files then also I get only default two file. Dynamically added file object are not available in controller.  
Please help...!!.

[Reply](#)**Venki**

19 July, 2013, 21:42

I am getting the java.io.IOException: Destination file [C:\TEMP] already exists and could not be deleted.  
However the file is getting uploaded and updated.  
Please see my code below :

```
try {
    if(multiPartFile !=null) {
        InputStream inputStream = multiPartFile.getInputStream();
        multiPartFile.transferTo( new File(SAVE_DIR + multiPartFile.getOriginalFilename()));
        deploymentStackMap.setUploadFileName( multiPartFile.getOriginalFilename().toString());
    }
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

[Reply](#)**prem**

25 July, 2013, 11:46

i want to show the image after upload image so can u help me to get that code in .jsp and also controller?

[Reply](#)**Devesh Anand**

22 August, 2013, 18:41

Hi viral, Really this examples helps me a lot, But plz do one help for me, plz develop a applicaiton to select two files from browser ( txt1, txt2) and display it on to the browser and while clicking on save button it will ask the location on harddisk to save the files.  
And plz send me these application on to the my id – [deveshanand2@gmail.com](mailto:deveshanand2@gmail.com)  
and if possible then give me ur number also into my id.  
Thanking you  
=====

Devesh Anand

[Reply](#)**jimmi**

3 October, 2013, 23:33

How many changes are require to convert this to Ajax based file upload?

[Reply](#)**Amuthan**

12 November, 2013, 13:13

I have repository class where I want to store the image that is coming from MultipartFile to webapp/resources/images directory before adding the product details to the repository ?

Spring MVC Multiple File Upload tutorial. Spring Multiple File upload example

```
@Override  
    public void addProduct(Product product) {  
        Resource resource = resourceLoader.getResource("file:webapp/resources/images");  
  
        MultipartFile proudctImage = product.getProudctImage();  
  
        try {  
            proudctImage.transferTo(new File(resource.getFile() + proudctImage.getOriginalFilename()));  
        } catch (IOException e) {  
            throw new RuntimeException(e);  
        }  
        listOfProducts.add(product);  
    }  
}
```

my repository class is ResourceLoaderAware. I am getting fileNotFoundException  
“imagesDesert.jpg” is the image I am trying to upload

java.io.FileNotFoundException: webapp\resources\imagesDesert.jpg (The system cannot find the path specified)

## Reply

santhosh

20 November, 2013, 11:09

Hi Viral,  
Do you have any idea for sending Email with multiple attachments???? actually i refereed below link <http://www.codejava.net/frameworks/spring/spring-mvc-send-e-mail-with-attachments> its working fine for single attachment...  
please if u have any idea share us...  
Thank you

Reply

phyd

28 January, 2011, 10:10

```
hey I want to remove attach files...
plz give me code as early as possible.
function confirmation()
{
alert("Please check your mail and attachment")
}
$(document).ready(function() {
//add more file components if Add is clicked
$('#addFile').click(function() {
var fileIndex = $('#fileTable tr').children().length;
if(fileIndex==0){
fileIndex=fileIndex;
}
else if(fileIndex==2){
fileIndex=fileIndex-1;
}
else{
var i=fileIndex/2-1;
fileIndex=i+1;
}
$('#fileTable').append(
'
<tr><td>' +
'<input type="fil
' + '</td>' +
'<td>' +
'<input id="delfi
' + '</td></tr>
');

});
$('#delfile').live('click', function() {
$(this).parent().parent().html("");
return false;
});
});
java code-----
```

```
public void sendingmail(mail mail) {  
  
    try{  
        MimeMessage mimemsg=javaMailSender.createMimeMessage();  
        MimeMessageHelper mmHelper=new MimeMessageHelper(mimemsg,true);  
        mmHelper.setFrom("mujum@xcalibertech.com");  
        mmHelper.setTo(mail.getTo());
```

## Spring MVC Multiple File Upload tutorial. Spring Multiple File upload example

```

mmHelper.setSubject(mail.getSub());
mmHelper.setText(mail.getText());

List<MultipartFile> files=mail.getFiles();
List<String> fileNames = new ArrayList<String>();
String fileName;
float total=0.0f;
float total1=0.0f;
float total2=0.0f;

if(null != files && files.size() > 0) {
    for (MultipartFile multipartFile : files) {

        fileName = multipartFile.getOriginalFilename();

        String path="D:/Attachments/"+fileName;
        File filenew=new File(path);
        float byt=filenew.length();
        total=total+byt;
        total1=total/1024;
        total2=total1/1024;

        try {
            multipartFile.transferTo(filenew);

        } catch (IllegalStateException e) {

            e.printStackTrace();
        } catch (IOException e) {

            e.printStackTrace();
        }
        String newpath=filenew.getAbsolutePath();
        System.out.println(filenew.getAbsolutePath());

        FileSystemResource file = new FileSystemResource(new File(newpath));

        mmHelper.addAttachment(fileName, file);
    }

    System.out.println("\ntotal size="+total2+" MB");

}
javaMailSender.send(mimemsg);

}catch(MessagingException e)
{
    e.printStackTrace();
}

}

```

thank you.....

[Reply](#)

### Leave a Reply

Your email address will not be published. Required fields are marked \*

Name \*

Email \*

Website

Comment

**Note**

To post source code in comment, use **[code language] [/code]** tag, for example:

- [code java] Java source code here [/code]
- [code html] HTML here [/code]

[Post Comment](#)

---

**Categories**

[Home page](#)   [Struts](#)   [Spring](#)   [AJAX](#)   [PHP](#)   [Java](#)   [JavaEE](#)  
[JavaScript](#)   [CSS](#)   [Database](#)   [Web 2.0](#)   [News](#)   [Fun](#)  
[General](#)   [Featured](#)   [Play Framework](#)   [Android](#)  
[FreeMarker Template](#)   [HTML5](#)

---

**Site Pages**

[About viralpatel.net](#)   [Join Us](#)   [Search](#)   [Advertise](#)  
[Posts Feed](#)   [Comments Feed](#)

[Back to top](#)

Copyright © 2014 ViralPatel.net. All rights reserved :)