

مثالی برای File Upload برای Spring MVC

در این مثال چگونگی استفاده از File Upload Control در فرم ها با استفاده از Spring Web MVC Framework نشان داده شده است. برای شروع Eclipse IDE را آماده کنید و جهت توسعه ی یک برنامه ی وب پویا با استفاده از Spring Web Framework مراحل زیر را دنبال کنید.

مرحله	توضیحات
1	پروژه ای با نام HelloWorld در بسته ی com.tutorialspoint همان طور که در بخش Spring MVC - Hello World توضیح داده شده است، ایجاد کنید.
2	در بسته ی com.tutorialspoint کلاس های جاوای FileModel و FileUploadController را ایجاد کنید.
3	در زیر پوشه ی jsp فایل های ویوی fileUpload.jsp و success.jsp را ایجاد کنید.
4	در زیرپوشه ی WebContent پوشه ی temp را ایجاد کنید.
5	کتابخانه ی commons-io.jar و commons-fileupload.jar را دانلود کنید و آن ها در CLASSPATH خود قرار دهید.
6	مرحله ی نهایی ایجاد محتوای فایل های منبع و پیکربندی و اکسپورت کردن برنامه به صورت زیر است.

FileModel.java

```
package com.tutorialspoint;

import org.springframework.web.multipart.MultipartFile;

public class FileModel {

    private MultipartFile file;
```

```
public MultipartFile getFile() {  
  
    return file;  
  
}  
  
public void setFile(MultipartFile file) {  
  
    this.file = file;  
  
}  
  
}
```

FileUploadController.java

```
package com.tutorialspoint;  
  
import java.io.File;  
import java.io.IOException;  
  
import javax.servlet.ServletContext;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.ModelMap;  
import org.springframework.util.FileCopyUtils;  
import org.springframework.validation.BindingResult;  
import org.springframework.validation.annotation.Validated;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.multipart.MultipartFile;  
import org.springframework.web.servlet.ModelAndView;
```

```
@Controller

public class FileUploadController {

    @Autowired

    ServletContext context;

    @RequestMapping(value = "/fileUploadPage", method = RequestMethod.GET)

    public ModelAndView fileUploadPage() {

        FileModel file = new FileModel();

        ModelAndView modelAndView = new ModelAndView("fileUpload", "command", file);

        return modelAndView;

    }

    @RequestMapping(value="/fileUploadPage", method = RequestMethod.POST)

    public String fileUpload(@Validated FileModel file, BindingResult result, ModelMap model) throws IOException {

        if (result.hasErrors()) {

            System.out.println("validation errors");

            return "fileUploadPage";

        } else {

            System.out.println("Fetching file");

            MultipartFile multipartFile = file.getFile();

            String uploadPath = context.getRealPath("") + File.separator + "temp" + File.separator;

            //Now do something with file...

            FileCopyUtils.copy(file.getFile().getBytes(), new File(uploadPath+file.getFile().getOriginalFilename()));

            String fileName = multipartFile.getOriginalFilename();

            model.addAttribute("fileName", fileName);

        }

    }

}
```

```
    return "success";
}
}
}
```

HelloWeb-servlet.xml

```
<beans xmlns = "http://www.springframework.org/schema/beans"
xmlns:context = "http://www.springframework.org/schema/context"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation = "
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.0.xsd">

<context:component-scan base-package = "com.tutorialspoint" />

<bean class = "org.springframework.web.servlet.view.InternalResourceViewResolver">
<property name = "prefix" value = "/WEB-INF/jsp/" />
<property name = "suffix" value = ".jsp" />
</bean>

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

در اینجا در اولین متد سرویس `fileUploadPage()` شیء `FileModel` خالی را در شیء `ModelAndView` با نام `"command"` عبور داده ایم. دلیل انجام این کار این است که `spring framework` انتظار دارد اسم شیء

"command" باشد. این برای حالتی است که ما از تگ های <form:form> در فایل JSP استفاده کنیم. بنابراین زمانی که متد () fileUploadPage فراخوانی می شود، این متد ویوی fileUpload.jsp را برگشت می دهد. دومین متد سرویس fileUpload() در برابر یک متد POST و در آدرس HelloWeb/fileUploadPage فراخوانی می شود. شما باید شیء مدل خود را بر اساس اطلاعات ارائه شده آماده کنید. در نهایت یک ویوی "success" از متد سرویس برگشت داده می شود. این امر باعث می شود success.jsp نمایش داده شود.

fileUpload.jsp

```
<%@ page contentType="text/html; charset = UTF-8" %>
<%@ taglib prefix = "form" uri = "http://www.springframework.org/tags/form"%>
<html>
<head>
<title>File Upload Example</title>
</head>
<body>
<form:form method = "POST" modelAttribute = "fileUpload"
enctype = "multipart/form-data">
Please select a file to upload :
<input type = "file" name = "file" />
<input type = "submit" value = "upload" />
</form:form>
</body>
</html>
```

در اینجا برای نگاشت file Upload control در server model از صفت modelAttribute در کنار value="fileUpload" استفاده کرده ایم.

success.jsp

```
<%@ page contentType = "text/html; charset = UTF-8" %>

<html>

<head>

<title>File Upload Example</title>

</head>

<body>

FileName :

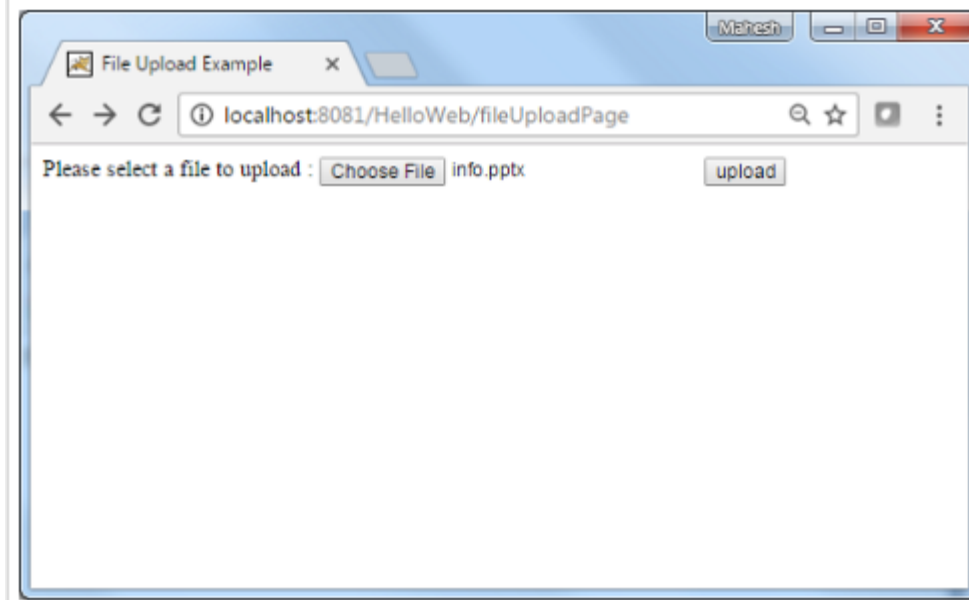
<b>${fileName}</b> - Uploaded Successfully.

</body>

</html>
```

بعد از آن که کار فایل های پیکربندی و منبع تمام شد، برنامه ی خود را اکسپورت کنید. بر روی برنامه ی خود کلیک راست کنید، از گزینه ی Export → WAR File استفاده کنید و فایل HelloWeb.war را داخل پوشه ی webapps متعلق به Tomcat ذخیره کنید.

حالا سرور Tomcat را اجرا کنید و مطمئن شوید که از طریق پوشه ی webapps و با استفاده از یک مرورگر استاندارد می توانید به دیگر صفحات وب دسترسی پیدا کنید. حالا در صورت وارد کردن آدرس <http://localhost:8080/HelloWeb/fileUploadPage> و در صورت نبود مشکل در برنامه ی Spring Web صفحه ی زیر نمایش داده می شود.



بعد از وارد کردن اطلاعات مورد نیاز، بر روی دکمه ی submit کلیک کنید تا اگر مشکلی در برنامه ی Spring Web وجود نداشته باشد، صفحه زیر نمایش داده شود.

