[13] FileUpload

1. Pom.xml에 의존추가

<!-- 파일업로드 -->

<dependency>

<groupId>commons-fileupload</groupId>

<artifactId>commons-fileupload</artifactId>

<version>1.5</version>

</dependency>

1. Web.xml 한글필터처리부분 추가

<!-- 한글처리 -->

<filter>

<filter-name>encodingFilter</filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

<init-param>

<param-name>encoding</param-name>

<param-value>UTF-8</param-value>

</init-param>

<init-param>

<param-name>forceEncoding</param-name>

<param-value>true</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>encodingFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

1. Servlet-context.xml (웹과 관련된 설정부분 여기다) cf.웹이외의 설정부분은 root-context.xml
   * multipartResolver Bean생성

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

<!-- maxUploadSize : 최대 업로드 가능한 바이트 크기

maxInMemorySize : 디스크에 임시파일을 생성하기 전에 메모리에 보관할 수 있는 최대 바이트 크기 1M의 경우1024\*1024=10485760,-1은 용량 제한 없음

defaultEncoding : 요청을 파싱할 때 사용할 캐릭터 인코딩. 기본은 ISO-8859-1 -->

<beans:property name=*"maxUploadSize"* value=*"10485760"*/>

</beans:bean>

1. Servlet-context.xml 에 추가

<resources mapping=*"/upload/\*\*"* location=*"/upload/"* />

1. view

* **file\_input.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<div align=*"center"*>

<hr width=*"500"* color=*"green"*>

<h2>파일 upload</h2>

<hr width=*"500"* color=*"green"*>

<form action=*"fileUpload"* method=*"post"* enctype=*"multipart/form-data"*>

<p>첨부파일1 : <input type=*"file"* name=*"file1"*></p>

<p>첨부파일2 : <input type=*"file"* name=*"file2"*></p>

<p>첨부파일3 : <input type=*"file"* name=*"file3"*></p>

<input type=*"submit"* value=*"전송"*>

</form>

</div>

</body>

</html>

* + **fileResult.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<h2>전송결과 : ${fileResult}</h2>

파일0 : upload/${filename[0] }<br>

파일1 : upload/${filename[1] }<br>

파일2 : upload/${filename[2] }<br>

<img src=*"upload/*${filename[0] }*"*>

<img src=*"upload/*${filename[1] }*"*>

<img src=*"upload/*${filename[2] }*"*>

</body>

</html>

1. Controller
   * FileController.java

@Controller

**public** **class** FileController {

@Autowired

**Private** FileUploadService fService;

@RequestMapping(value="fileInput", method = RequestMethod.***GET***)

**public** String fileInput() {

**return** "fileInput";

}

@RequestMapping(value="fileUpload", method = RequestMethod.***POST***)

**public** ModelAndView fileupload(MultipartHttpServletRequest mRequest, ModelAndView mav) {

**if**(fService.fileUp(mRequest, mav)) {

mav.addObject("fileResult","파일 업로드 성공");

}**else** {

mav.addObject("fileResult","파일 업로드 실패");

}

mav.setViewName("fileResult");

**return** mav;

}

}

1. Service
   * Upload.java

@Service

**public** **class** FileUploadService {

**public** **boolean** fileUp(MultipartHttpServletRequest mRequest, ModelAndView mav) {

**boolean** isUpload = **false**;

String uploadPath = mRequest.~~getRealPath~~("upload/");

String backupPath = "D:/webPro/source/9\_Spring/ch13\_fileup/src/main/webapp/upload/";

String[] fileNames = **new** String[3];

**int** i = 0;

Iterator<String> params = mRequest.getFileNames(); // file1, file2, file3

**while**(params.hasNext()) {

String param = params.next();

System.***out***.println(i+"번째 파라미터 : "+param);

MultipartFile mFile = mRequest.getFile(param); // 파라미터의 파일 객체

String originalFileName = mFile.getOriginalFilename(); // 업로드 했을 때 원래 파일 이름

fileNames[i] = originalFileName;

System.***out***.println(originalFileName==**null**? "null이야":fileNames[i].equals("")? "빈칸이야":"빈칸이 아냐");

// 첨부를 안 하면 빈칸

**if**(fileNames[i]!=**null** && !fileNames[i].equals("")) { // 첨부함

**if**(**new** File(uploadPath+fileNames[i]).exists()) {

// 첨부파일과 같은 이름의 파일이 서버에 존재하는 경우 -> 파일이름을 변경

fileNames[i] = System.*currentTimeMillis*() + fileNames[i];

}// if - 파일이름 변경

**try** {

mFile.transferTo(**new** File(uploadPath+fileNames[i]));// 서버에 파일을 저장하는 로직

System.***out***.println("서버에 저장된 파일 : "+uploadPath+fileNames[i]);

System.***out***.println("백업위해 복사할 파일:" +backupPath+fileNames[i]);

isUpload = filecopy(uploadPath+fileNames[i], backupPath+fileNames[i]);//backup

} **catch** (IOException e) {

System.***out***.println(e.getMessage());

}

}//if

i++;

}// while

mav.addObject("fileNames", fileNames);

**return** isUpload;

}

**private** **boolean** filecopy(String serverFile, String backupFile) {

**boolean** isCopy = **false**;

FileInputStream is = **null**;

FileOutputStream os = **null**;

**try** {

File file = **new** File(serverFile);

is = **new** FileInputStream(serverFile);

os = **new** FileOutputStream(backupFile);

**byte**[] buff = **new** **byte**[(**int**)file.length()];

**while**(**true**) {

**int** nReadByte = is.read(buff);

**if**(nReadByte == -1) **break**;

os.write(buff, 0, nReadByte);

}

isCopy = **true**;

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

}**finally** {

**try** {

**if**(os!=**null**) os.close();

**if**(is!=**null**) is.close();

} **catch** (Exception e) {

System.***out***.println(e.getMessage());

}

}

**return** isCopy;

}

}