

Spring MVC : Controller 파일업로드처리

🕒 생성일	@2022년 8월 20일 오후 9:33
🏷 태그	

Spring Legacy Project로 생성되는 경우, Servlet 2.5를 기준으로 생성되기 때문에, 이 기준으로 파일업로드 처리가 필요하다.

1. pom.xml

commons-fileupload 라이브러리 추가

```
<dependency>
  <groupId>commons-fileupload</groupId>
  <artifactId>commons-fileupload</artifactId>
  <version>1.4</version>
</dependency>
```

2. servlet-context.xml

multipartResolver Bean 객체를 등록해주어야한다. **id를 반드시, multipartResolver로 정확히 설정해주어야 한다.**

속성	설명
defaultEncoding	업로드하는 파일의 이름이 한글일 경우, 깨짐을 방지한다.
maxUploadSize	한 번의 Request로 전달 될 수 있는 최대 크기 (bytes)
maxUploadSizePerFile	파일 하나의 최대 크기 (bytes)
maxInMemorySize	디스크에 저장하지 않고 메모리에 유지하는 최대 크기
preserveFilename	임시파일명을 쓰지 않고, 원본파일을 유지하지하기 위한 속성
uploadTempDir	업로드된 파일이 저장되는 임시폴더

```

<beans:bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <beans:property name="defaultEncoding" value="utf-8" />
    <!-- 1024 * 1024 * 10 bytes 10MB -->
    <beans:property name="maxUploadSize" value="104857560"/>
    <!-- 1024 * 1024 * 2 bytes 2MB -->
    <beans:property name="maxUploadSizePerFile" value="2097152"/>
    <beans:property name="maxInMemorySize" value="10485756"/>
    <beans:property name="preserveFilename" value="true"/>
    <beans:property name="uploadTempDir" value="file:///C:/Temp"/>
</beans:bean>

```

3. jsp 작성

- . form 태그로 action은 /fileupload/doit으로 전송 (HTTP method는 POST)
- . 전송되는 file의 name은 files로 @RequestParam("files")로 작성 및 여러개 전송되므로, List 객체로 받아야 한다.

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>File Upload</title>
</head>
<body>
    <h1>/WEB-INF/views/fileupload/page.jsp</h1>
    <hr>

    <form action="/fileupload/doit" method="POST" enctype="multipart/form-data">
        <div>Name : <input type="text" name="myName" size="10"></div>
        <div>Age : <input type="text" name="myAge" size="5"></div>

        <div><input type="file" name="files"></div>
        <div><input type="file" name="files"></div>
        <div><input type="file" name="files"></div>
        <div><input type="file" name="files"></div>
        <div><input type="file" name="files"></div>

        <br>

        <input type="submit" value="Upload">
    </form>

```

```
</body>
</html>
```

4. Controller 작성

/fileupload/page 로 접속 시, 3에서 작성한 view가 보이도록 한다.

```
@GetMapping("page")
public void fileUploadPage() {
    log.trace("fileUploadPage() invoked.");

    // void리턴타입으로 view는 /fileupload/page로 이동된다.
} // fileUploadPage
```

임시폴더로 파일을 업로드하는 핸들러 작성

- 파일 전송이 다중으로 들어올 예정으로, List 객체로 업로드할 파일을 수집한다.
- MultipartFile의 transferTo 메소드를 통해, 저장을 할 폴더로 임시파일을 저장한다.

```
@PostMapping("/doit")
public void DoFileUpload(String myName, String myAge,
    @RequestParam("files") List<MultipartFile> files
) {
    log.trace("DoFileUpload({}, {}, files) invoked.", myName, myAge);

    String targetDir = "C:/Temp/upload/";

    for(MultipartFile file : files) {
        log.info("name : {}", file.getOriginalFilename());
        log.info("size : {}", file.getSize());

        // 파일이 저장되는 위치
        String targetFile = targetDir + file.getOriginalFilename();

        try {
            file.transferTo(new File(targetFile));
        } catch (IllegalStateException | IOException e) {
            e.printStackTrace();
        } // try-catch

    } // enhanced for

} // DoFileUpload
```

Clinet에서 파일업로드 과정

-. fileupload/page.jsp에서 파일을 전송

/WEB-INF/views/fileupload/page.jsp

Name :

Age :

파일 선택	html.png
파일 선택	java.png
파일 선택	Backend.png
파일 선택	algorithm.png
파일 선택	cs.png

-. Controller 핸들러에 의해, 수집한 파일의 정보

```
22:24:38.721 TRACE --- [http-nio-8080-exec-3] o.z.m.c.FileUploadController.DoFileUpload:35 - DoFileUpload(Ryan, 23, files) invoked.
22:24:38.722 INFO --- [http-nio-8080-exec-3] o.z.m.c.FileUploadController.DoFileUpload:40 - name : algorithm.png
22:24:38.722 INFO --- [http-nio-8080-exec-3] o.z.m.c.FileUploadController.DoFileUpload:41 - size : 11127
22:24:38.723 INFO --- [http-nio-8080-exec-3] o.z.m.c.FileUploadController.DoFileUpload:40 - name : Backend.png
22:24:38.723 INFO --- [http-nio-8080-exec-3] o.z.m.c.FileUploadController.DoFileUpload:41 - size : 15045
22:24:38.724 INFO --- [http-nio-8080-exec-3] o.z.m.c.FileUploadController.DoFileUpload:40 - name : cs.png
22:24:38.724 INFO --- [http-nio-8080-exec-3] o.z.m.c.FileUploadController.DoFileUpload:41 - size : 15976
22:24:38.725 INFO --- [http-nio-8080-exec-3] o.z.m.c.FileUploadController.DoFileUpload:40 - name : java.png
22:24:38.725 INFO --- [http-nio-8080-exec-3] o.z.m.c.FileUploadController.DoFileUpload:41 - size : 24092
22:24:38.726 INFO --- [http-nio-8080-exec-3] o.z.m.c.FileUploadController.DoFileUpload:40 - name : html.png
22:24:38.726 INFO --- [http-nio-8080-exec-3] o.z.m.c.FileUploadController.DoFileUpload:41 - size : 30652
```

-. Controller 핸들러에 의해, 수집한 파일이 임시폴더에 저장

→ 저장위치는 "C:/Temp/upload/" + "file의 originalFilename" 이다.

