

### 3. 파일업로드



# 차례

---

1. Spring boot 파일업로드 환경 설정
2. 파일업로드 구현
3. 첨부파일 조회

# 1. 환경설정

---

- application.properties

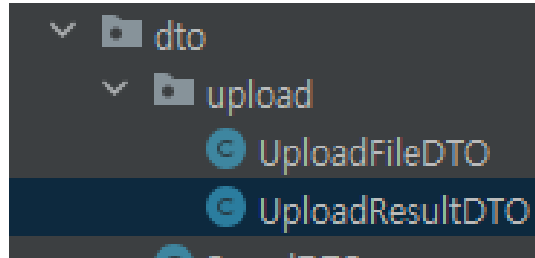
```
spring.servlet.multipart.enabled=true  
spring.servlet.multipart.location=d:\\upload  
spring.servlet.multipart.max-request-size=30MB  
spring.servlet.multipart.max-file-size=10MB  
  
com.example.japb01.upload.path=d:\\upload # controller에서 사용할 upload path value
```

- build.gradle에 의존성 추가

```
// https://mvnrepository.com/artifact/net.coobird/thumbnailator  
implementation group: 'net.coobird', name: 'thumbnailator', version: '0.4.20'
```

## 2. 파일업로드 구현(form 방식)

- UploadFileDTO와 UploadResultDTO 작성



```
@Data
public class UploadFileDTO {
    private List<MultipartFile> files;
}
```

```
@Data
@Builder
@AllArgsConstructor
@NoArgsConstructor
public class UploadResultDTO {

    private String uuid;
    private String filename;
    private boolean image;

    public String getLink(){
        if(image){
            return "s_"+uuid+"_"+filename;
        }else{
            return uuid+"_"+filename;
        }
    }
}
```

## 2. 파일업로드 구현(form 방식)

---

### ■ controller 구현

```
@Controller
@Log4j2
@RequestMapping("/upload")
public class UpDownController {

    // application.properties에 설정된 upload path
    @Value("${com.example.japb01.upload.path}") // import 시에 springframework으로 시작하는 Value
    private String uploadPath;

    ...
}
```

## 2. 파일업로드 구현(form 방식)

### ■ controller 구현(계속)

```
@GetMapping("/uploadForm")
public void uploadForm(){
}
```

```
@PostMapping(value = "/uploadPro", consumes = MediaType.MULTIPART_FORM_DATA_VALUE)
public void upload(UploadFileDTO uploadFileDTO){

    log.info("~~~~~"+uploadFileDTO);

    if(uploadFileDTO.GetFiles() != null) {

        uploadFileDTO.GetFiles().forEach(multipartFile -> {
            String originalName=multipartFile.getOriginalFilename();
            log.info(originalName);
            String uuid= UUID.randomUUID().toString();
            Path savePath= Paths.get(uploadPath, uuid+"_"+originalName);
            boolean image=false;

            try{
                multipartFile.transferTo(savePath);
                if(Files.probeContentType(savePath).startsWith("image")){
                    image = true;
                    File thumbFile = new File(uploadPath, "s_" + uuid+"_"+ originalName);
                    Thumbnailator.createThumbnail(savePath.toFile(), thumbFile, 200,200);
                }
            }catch (IOException e){ e.printStackTrace(); }
        });
    }
}
```

## 2. 파일업로드 구현(form 방식)

---

### ■ 첨부파일 조회(controller)

```
@GetMapping("/view/{fileName}")
public ResponseEntity<Resource> viewFileGET(@PathVariable String fileName){

    Resource resource = new FileSystemResource(uploadPath+File.separator + fileName);
    String resourceName = resource.getFilename();
    HttpHeaders headers = new HttpHeaders();

    try{
        headers.add("Content-Type", Files.probeContentType( resource.getFile().toPath() ));
    } catch(Exception e){
        return ResponseEntity.internalServerError().build();
    }
    return ResponseEntity.ok().headers(headers).body(resource);
}
```

## 2. 파일업로드 구현(form 방식)

### ■ 첨부파일 조회(화면구현)

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
      layout:decorate="~{layout/basic.html}">

  <div layout:fragment="content">
    <div class="col">
      <div>[[${fileList}]]</div>
      <div class="card" th:if="${fileList != null && fileList.size() > 0}">
        <div th:each="fileDTO:${fileList}">
          <div th:if="${fileDTO.image}">
            
          </div>
          <div th:unless="${fileDTO.image}">
            
            <span th:each="fileDTO:${fileList}">[[${fileDTO.getLink()}]]</span>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```



## 2. 파일업로드 구현(form 방식)

### ■ 파일삭제 구현

```
@GetMapping("/remove/{fileName}")
public String removeFile(@PathVariable String fileName){

    Resource resource = new FileSystemResource(uploadPath+File.separator + fileName);
    String resourceName = resource.getFilename();

    Map<String, Boolean> resultMap = new HashMap<>();
    boolean removed = false;

    try {
        String contentType = Files.probeContentType(resource.getFile().toPath());
        removed = resource.getFile().delete();

        //섬네일이 존재한다면
        if(contentType.startsWith("image")){
            File thumbnailFile = new File(uploadPath+File.separator+"s_" + fileName);
            thumbnailFile.delete();
        }
    } catch (Exception e) {
        log.error(e.getMessage());
    }
    return "redirect:/upload/uploadForm";
}
```