**File Upload and Download in SpringBoot-2022**

**Maven Pom.xml (SpringBoot Version 2.6.9)**

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.6.9</version>

<relativePath />

</parent>

**Controller Layer (Upload a document along with json data)**

@PostMapping(path = "/resume", consumes = { MediaType.***MULTIPART\_FORM\_DATA\_VALUE*** })

**public** ResponseEntity<String> uploadResume(

**@RequestPart("profileAsString") String profileStr,**

**@RequestPart("file") MultipartFile document**) {

**try** {

ObjectMapper mapper = JsonMapper.*builder*()

.enable(JsonReadFeature.***ALLOW\_UNQUOTED\_FIELD\_NAMES***).build();

Profile profile = mapper.readValue(profileStr, Profile.**class**);

System.***out***.println("Now Profile: " + profile);

storeResume(document);

} **catch** (JsonProcessingException e) {

e.printStackTrace();

}

System.***out***.println(document.getOriginalFilename());

**return** **new** ResponseEntity<>("Your resume submitted...", HttpStatus.***CREATED***);

}

**How to store the file**

**private** **void** storeResume(MultipartFile document) {

String dir = Paths.*get*("").toFile().getAbsolutePath();

// Create a directory called resumes

File resumeDir = **new** File(dir + File.***separator*** + "resumes");

**if** (!resumeDir.exists()) {

resumeDir.mkdirs();

}

String fileName = document.getOriginalFilename();

System.***out***.println("Uploaded Document File Name:" + fileName);

File file = **new** File(resumeDir.getAbsolutePath() + File.***separator*** + fileName);

**try** (OutputStream os = **new** FileOutputStream(file)) {

os.write(document.getBytes());

} **catch** (IOException e) {

e.printStackTrace();

}

System.***out***.println("Resume stored successfully");

}

**How to download a file or show a file in browser**

@RestController

**public** **class** FileDownloadController {

**private** **final** String BASE\_PATH = "E:/sure-delete";

@GetMapping("/download/{fileName:.+}")

**public** ResponseEntity<Resource> downloadFile(@PathVariable String fileName) {

Resource resource = getFileResource(fileName);

**String headerAttachement = "attachment; filename=\"" + resource.getFilename() + "\"";**

**return** ResponseEntity.*ok*().contentType(MediaType.***APPLICATION\_OCTET\_STREAM***)

.header(HttpHeaders.***CONTENT\_DISPOSITION***, headerAttachement)

.body(resource);

}

@GetMapping("/showfile/{fileName}")

**public** ResponseEntity<Resource> showFileInBrowser(@PathVariable String fileName) {

Resource resource = getFileResource(fileName);

MediaType fileType = fileName.endsWith("pdf") ? MediaType.***APPLICATION\_PDF*** : MediaType.***IMAGE\_JPEG***;

**String headerInline = "inline; filename=\"" + resource.getFilename() + "\"";**

**return** ResponseEntity.*ok*().contentType(fileType)

.header(HttpHeaders.***CONTENT\_DISPOSITION***, headerInline)

.body(resource);

}

**private** Resource getFileResource(String fileName) {

Path path = Paths.*get*(BASE\_PATH + File.***separator*** + fileName);

Resource resource = **null**;

**try** {

// resource = new InputStreamResource(new FileInputStream(new File(BASE\_PATH + File.separator + fileName)));

**resource = new UrlResource(path.toUri());**

} **catch** (IOException e) {

e.printStackTrace();

}

**return** resource;

}

}

**Few Notes**

**\*\* contentType(MediaType.*APPLICATION\_OCTET\_STREAM*) always forces the browser to download.**

**To download the file, always use this.**

**"attachment; filename=\"" + resource.getFilename() + "\"";**

**To show the file in browser, always use this**

**"inline; filename=\"" + resource.getFilename() + "\"";**