

File Upload Functionality in Spring Boot

POST http://localhost:8080/file/uploadFile Params Send Save

Authorization Headers Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary

Key	Value	Description	...	Bulk Ed
<input checked="" type="checkbox"/> file	<input type="button" value="Choose Files"/> SpringBootJBossRes...mplateSwagger.pdf	hello		
New key	Value	Description		

Body Cookies Headers (3) Test Results Status: 200 OK Time: 475 m

Pretty Raw Preview Text

```
1 SpringBootJBossRestTemplateSwagger.pdf :: File Received ::http://localhost:8080/downloadFile/SpringBootJBossRestTemplateSwagger.pdf
```

```
@SpringBootApplication(scanBasePackages="com.*")
public class TestApplication {

    public static void main(String[] args) {
        SpringApplication.run(TestApplication.class, args);
    }

}
```

```
@RestController
@RequestMapping("/file")
public class FileHandleController {

    @Autowired
    private Environment env;

    @PostMapping("/uploadFile")
    public String getFile(@RequestParam("file") MultipartFile file) throws Exception {
        Path fileStorageLocation = Paths.get(env.getProperty("file.upload-dir")).toAbsolutePath().normalize();

        Files.createDirectories(fileStorageLocation);
        String fileName = StringUtils.cleanPath(file.getOriginalFilename());

        Path targetLocation = fileStorageLocation.resolve(fileName);
        Files.copy(file.getInputStream(), targetLocation,
StandardCopyOption.REPLACE_EXISTING);

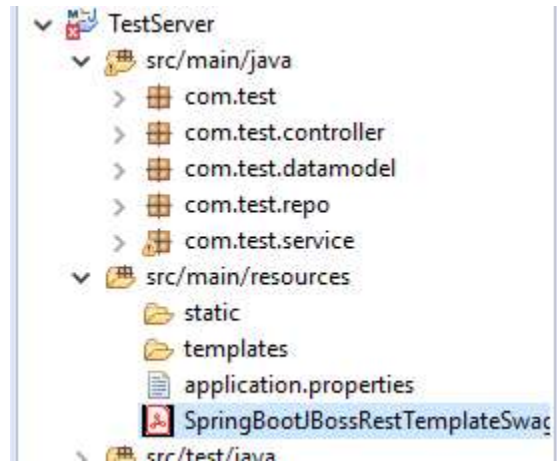
        String fileDownloadUri =
ServletUriComponentsBuilder.fromCurrentContextPath().path("/downloadFile/")
                .path(fileName).toUriString();
        return fileName + " :: File Received ::" + fileDownloadUri;
    }

}
```

Application.properties

```
spring.servlet.multipart.enabled=true  
spring.servlet.multipart.file-size-threshold=2KB  
spring.servlet.multipart.max-file-size=200MB  
spring.servlet.multipart.max-request-size=215MB
```

```
file.upload-dir=C:/Users/34060/Desktop/MyDocuments/Test/src/main/resources
```



ClientPart

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
```

upload.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <body>

    <h1>Spring Boot file upload example</h1>

    <form method="POST" action="/upload" enctype="multipart/form-data">
      <input type="file" name="file" /><br/><br/>
      <input type="submit" value="Submit" />
    </form>

  </body>
</html>
```

Spring Boot file upload example

Confirmation.pdf

Success Page

upload-success.html

File Uploaded successfully

```

@Controller
public class TestFileHandlerClient {

    @Autowired
    private RestTemplate restTemplate;

    @GetMapping("/")
    public String index() {
        return "upload";
    }

    @PostMapping(value = "/upload")
    public String sendFile(@RequestParam("file") MultipartFile file) {
        try {

            MultiValueMap<String, Object> parts = new LinkedMultiValueMap<String, Object>();
            parts.add("file", getUserFileResource(file));

            HttpHeaders headers = new HttpHeaders();
            headers.setContentType(MediaType.MULTIPART_FORM_DATA);
            HttpEntity<MultiValueMap<String, Object>> requestEntity = new HttpEntity
            <MultiValueMap<String, Object>>(parts, headers);

            ResponseEntity<String> response =
                restTemplate.exchange("http://localhost:8080/file/uploadFile",
                    HttpMethod.POST, requestEntity, String.class);

        } catch (Exception e) {
            System.out.println(e);
        }

        return "upload-success";
    }

    public static Resource getUserFileResource(MultipartFile fileIn) throws IOException
    {
        Path tempFile = Files.createTempFile(fileIn.getOriginalFilename(), ".pdf");
        Files.write(tempFile, fileIn.getBytes());
        File file = tempFile.toFile();
        return new FileSystemResource(file);
    }
}

```