



```

public class MainController {

    @RequestMapping(value = "/")
    public String index() {
        return "index.html";
    }

    @RequestMapping(value = "/upload", method = RequestMethod.POST, consumes =
MediaType.MULTIPART_FORM_DATA_VALUE)
    public String fileUpload(@RequestParam("file") MultipartFile file) {
        String result = "File was uploaded successfully";

        try {
            File convertFile = new File("/var/tmp/"+file.getOriginalFilename());
            convertFile.createNewFile();
            FileOutputStream fout = new FileOutputStream(convertFile);
            fout.write(file.getBytes());
            fout.close();

        } catch (IOException iex) {
            result = "Error " + iex.getMessage();
        } finally {
            return result;
        }
    }

    @RequestMapping(value = "/download", method = RequestMethod.GET)
    public ResponseEntity<Object> downloadFile() throws IOException {
        String fileName = "static/dump.txt";
        ClassLoader classLoader = new MainController().getClass().getClassLoader();

        File file = new File(classLoader.getResource(fileName).getFile());

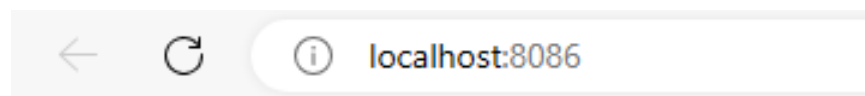
        InputStreamResource resource = new InputStreamResource(new
FileInputStream(file));
        HttpHeaders headers = new HttpHeaders();

        headers.add("Content-Disposition", String.format("attachment;
filename=\"%s\"", file.getName()));
        headers.add("Cache-Control", "no-cache, no-store, must-revalidate");
        headers.add("Pragma", "no-cache");
        headers.add("Expires", "0");

        ResponseEntity<Object>
responseEntity =
ResponseEntity.ok().headers(headers).contentType(
MediaType.parseMediaType("application/txt")).body(resource);

        return responseEntity;
    }
}

```




Upload file  No file chosen



Upload file  dump.txt



 dump.txt  
[Open file](#)