

Create a project to upload and download a file in Spring Boot.

index.html

```
<!DOCTYPE html>
<html>
<head><title>File Upload</title></head>
<body>

<form method="POST" enctype="multipart/form-data" action="/upload">
Upload file 
<input type="file" name="file" id="fileToUpload"><br><br>
<input type="submit" value="Upload " name="submit">
</form>
</body>
</html>
```

MainController.java

```
package com.ecommerce.controllers;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

import org.springframework.core.io.InputStreamResource;
import org.springframework.http.HttpHeaders;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.multipart.MultipartFile;

@Controller
public class MainController {
```

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

```
private static final String updir = "D:/var/tmp/";
```

```
@PostMapping(value = "/upload", consumes =
MediaType.MULTIPART_FORM_DATA_VALUE)
@ResponseBody
public String fileUpload(@RequestParam("file") MultipartFile file) {
    String result = "File was uploaded successfully";
    try {
        File directory = new File(updir);
        if (!directory.exists()) {
            directory.mkdirs();
        }

```

```
File convertFile = new File(updir + file.getOriginalFilename());
convertFile.createNewFile();
FileOutputStream fout = new FileOutputStream(convertFile);
fout.write(file.getBytes());
fout.close();
return result;
} catch (IOException iex) {
    result = "Error " + iex.getMessage();

```

```
} catch (Exception ex) {
    result = "Unexpected error: " + ex.getMessage();
}
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());
```

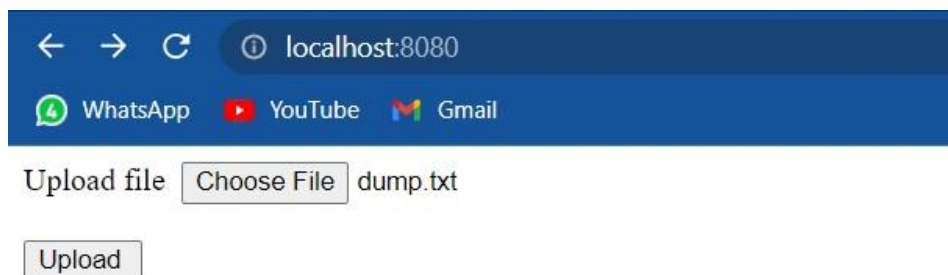
```
InputStreamResource resource = new InputStreamResource(new FileInputStream(file));  
HttpHeaders headers = new HttpHeaders();
```

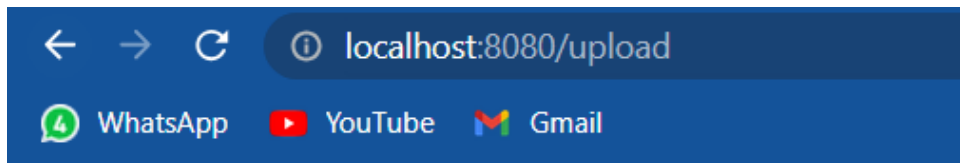
```
headers.add("Content-Disposition", String.format("attachment; filename=\"%s\"",  
file.getName()));  
headers.add("Cache-Control", "no-cache, no-store, must-revalidate");  
headers.add("Pragma", "no-cache");  
headers.add("Expires", "0");
```

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

```
return responseEntity;  
}  
}
```

OUTPUT





File was uploaded successfully

