

3.File upload and download

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.ClassPathResource;
import org.springframework.core.io.InputStreamResource;
import org.springframework.core.io.Resource;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Controller;
import org.springframework.util.ResourceUtils;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
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).contentTypeLength(file.length()).contentType(
    MediaType.parseMediaType("application/txt")).body(resource);

    return responseEntity;
}
}

```

index.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="file"><br><br>
        <input type="submit" value="Upload " name="submit">
    </form>
</body>
</html>

```

dump.txt:

This is a file which has been downloaded via Spring Boot

