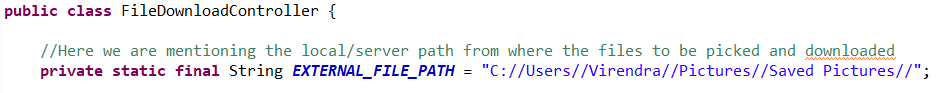
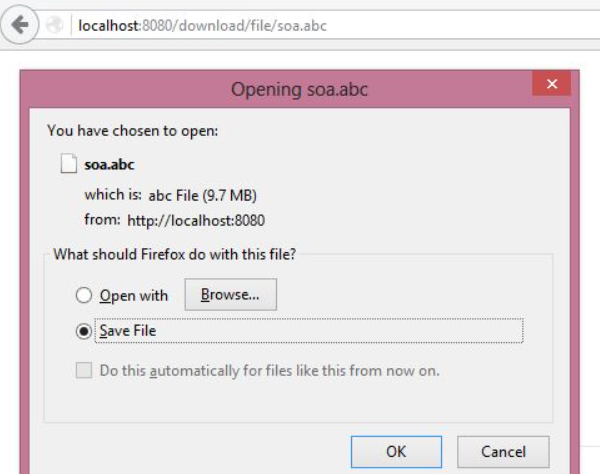


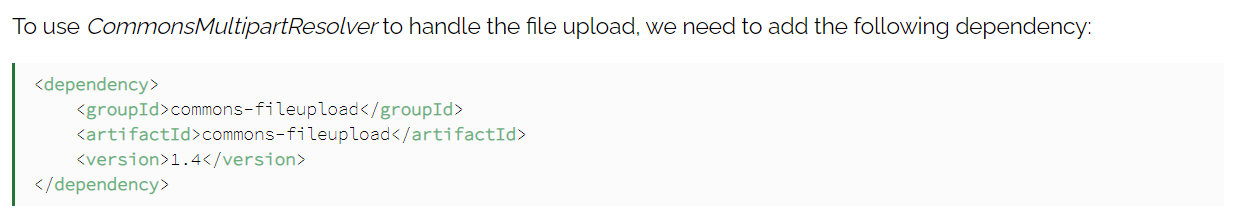
**File uploading:   
Steps to run  
1. Maven clean  
2. Maven install  
3. Run as “Run on server”  
4. Open the URL(in chrome browser) :** <http://localhost:8080/SpringFileUpload/upload.jsp>   
**The files will be uploaded using the JSP and saved into this location:**   
C:\Users\Virendra\Downloads\eclipse-jee-2020-12-R-win32-x86\_64\eclipse\null\tmpFiles

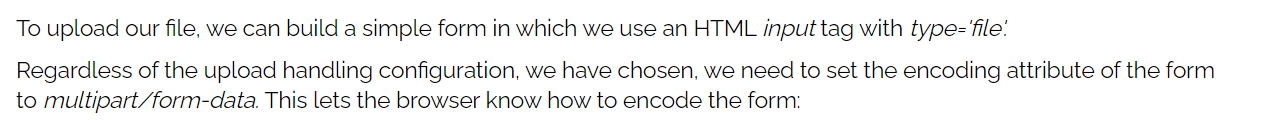
**File downloading:   
URL :** [http://localhost:8080/SpringFileUpload/download/file/<filename](http://localhost:8080/SpringFileUpload/download/file/%3cfilename)>   
Here we have to mention the path inside the code where the files will be picked and downloaded into your system.  
means; the “C://Users//Virendra//Pictures//Saved Pictures//” from here it will pick the files to be downloaded.  


The below line is used to show that file when called using the url mentioned above then the file will not be shown inside the browser or on web page but the file will be shown as  
//if you comment the below line then content will be shown in the browser only.

response.setHeader("Content-Disposition", String.*format*("attachment; filename=\"" + file.getName() + "\""));  


**File Upload:**

**Step 1:**   


**Step 2:**  
Now we can define the *CommonsMultipartResolver* bean into our Spring configuration  
  
  
**Step 3:**  


**Step 4:**Create a controller class to handle the request coming from the JSP Page.   
**To store the uploaded file we can use a***MultipartFile***variable.** We can retrieve this variable from the request parameter inside our controller's method:

