7. File Upload and Download in Postman

File Upload and Download

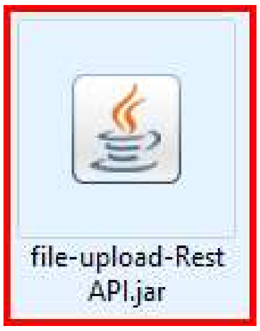
- ➔ When we upload a file (single or multiple) through a web UI, an API is triggered in the background.
- ➔ This API receives the file from the client and sends the request to the server. Then, the uploaded files are stored on the server or in the database.
- ➔ When you download a file, your browser asks the server for it. If the file is ready, the server says **"OK"** and sends the file. It also sends a **Content-Disposition message** that tells the browser, "This is a file to save," so the browser opens a save dialog or saves it automatically

Note

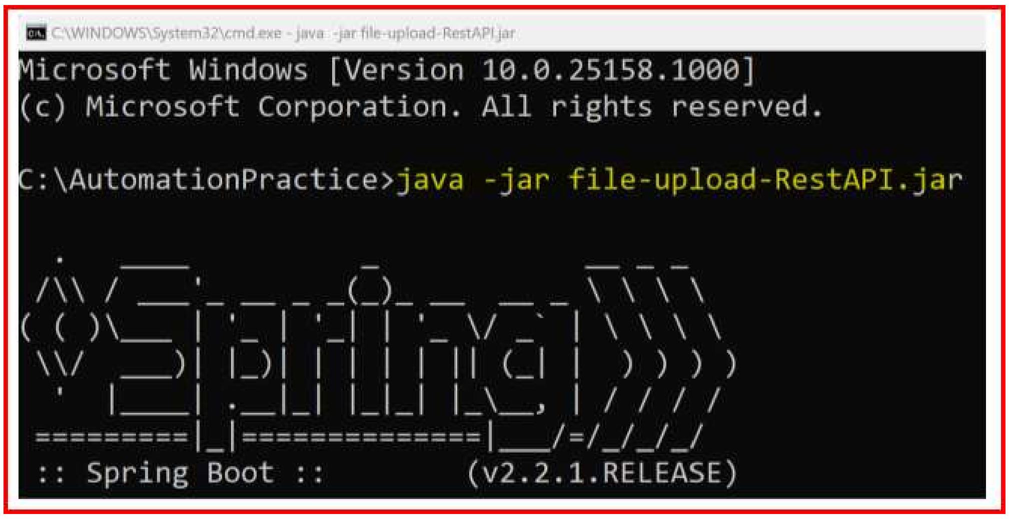
- ➔ It's hard to find public APIs for file uploads (single or multiple), so we will create our own API and test it using the Postman tool.

Setup API

- ➔ Download jar file from the link
  - ◆ https://github.com/Madhan-091296/spring-boot-file-upload-download-rest-api-master/blob/master/file-upload-RestAPI.jar



- ➔ Place below jar file in any location.
- ➔ Navigate to Jar file location then run below command in command prompt.
  - ◆ **java -jar file-upload-RestAPI.jar**



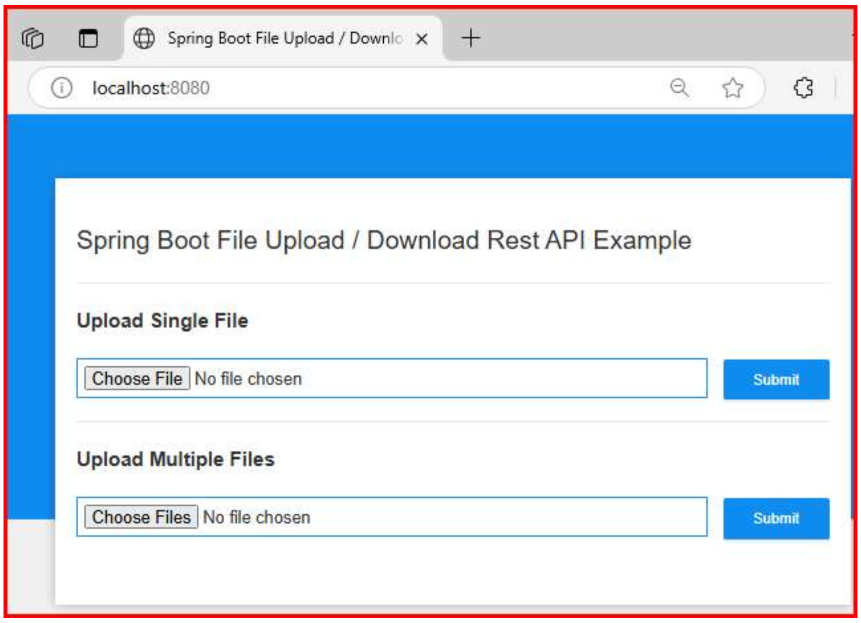
```
C:\WINDOWS\System32\cmd.exe - java -jar file-upload-RestAPI.jar
Microsoft Windows [Version 10.0.25158.1000]
(c) Microsoft Corporation. All rights reserved.

C:\AutomationPractice>java -jar file-upload-RestAPI.jar

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.2.1.RELEASE)
```

➔ Open below URL on browser then check working fine.

- ◆ http://localhost:8080/



Spring Boot File Upload / Download Rest API Example

Upload Single File

Choose File No file chosen Submit

Upload Multiple Files

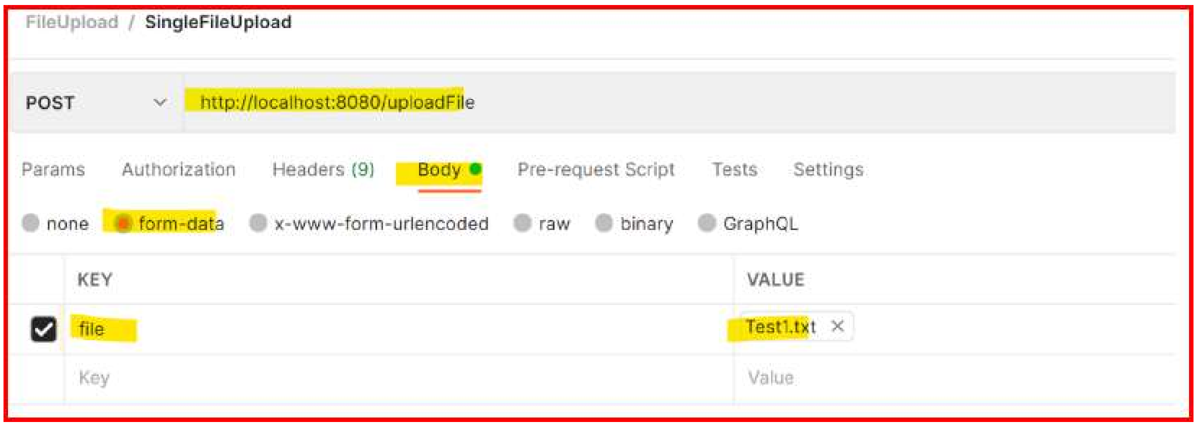
Choose Files No file chosen Submit

Workout in Postman

➔ Create **Day7_FileUpload** Collection in myworkspace

➔ **Upload Single File**

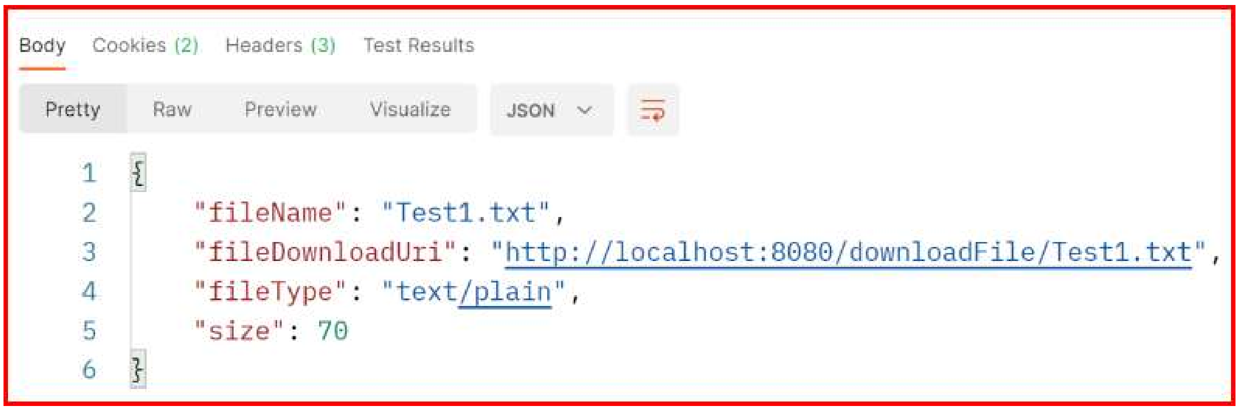
- ◆ Create a Post request **SingleFileUpload**
  - **POST:** http://localhost:8080/uploadFile



POST http://localhost:8080/uploadFile

Body: form-data

KEY	VALUE
✓ file	Test1.txt
Key	Value

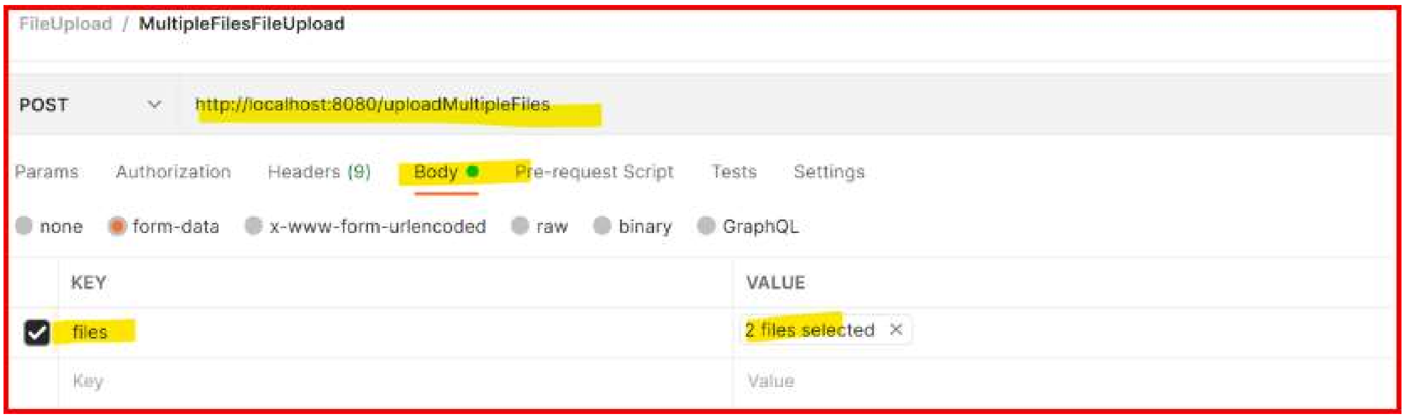


```
{
    "fileName": "Test1.txt",
    "fileDownloadUri": "http://localhost:8080/downloadFile/Test1.txt",
    "fileType": "text/plain",
    "size": 70
}
```

➔ **Upload Multiple Files**

◆ Create a Post request **MultipleFilesUpload**

- **POST:** http://localhost:8080/uploadMultipleFiles

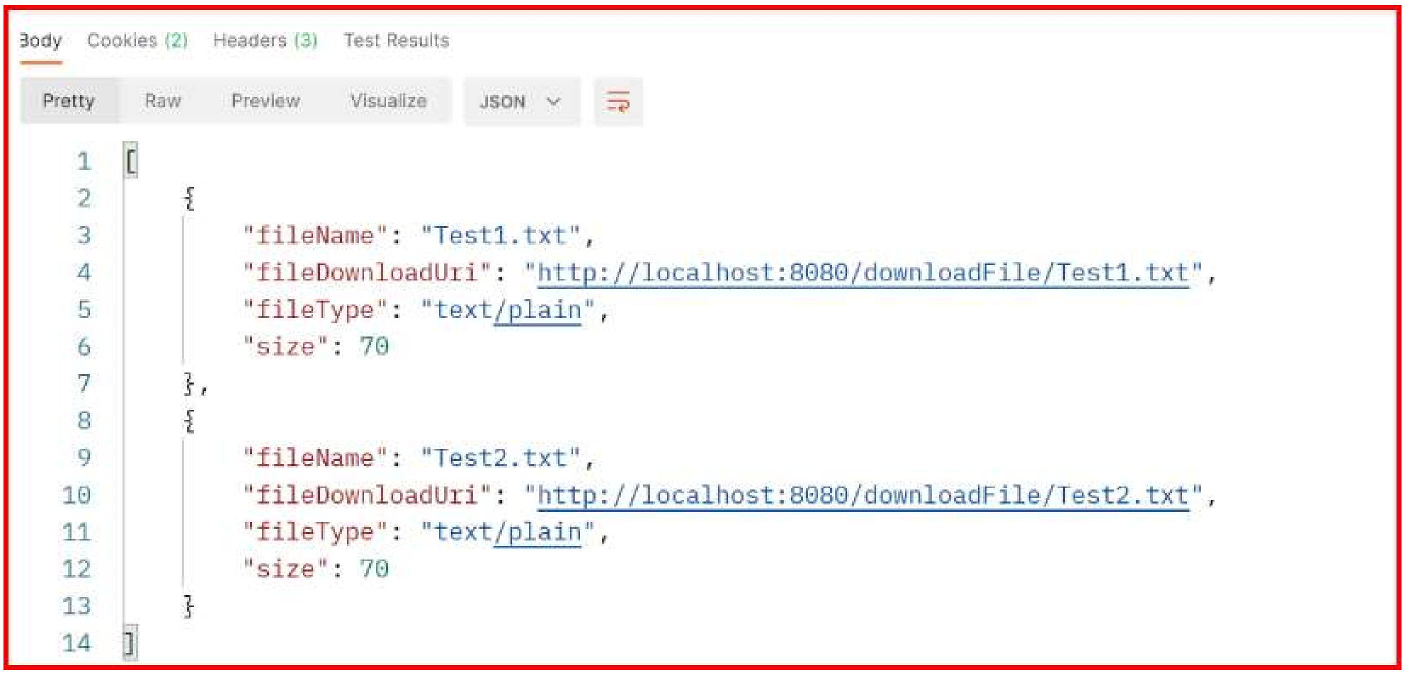

FileUpload / MultipleFilesFileUpload

POST http://localhost:8080/uploadMultipleFiles

Params | Authorization | Headers (9) | Body | Pre-request Script | Tests | Settings

none [✓ form-data] x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE
✓	files	2 files selected
	Key	Value


Body | Cookies (2) | Headers (3) | Test Results

Pretty | Raw | Preview | Visualize | JSON

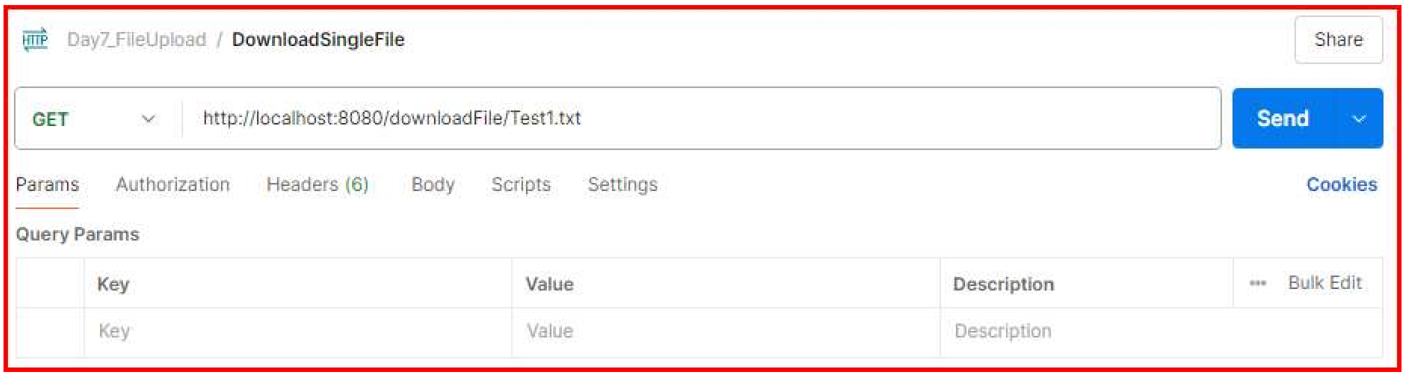
```
[
    {
        "fileName": "Test1.txt",
        "fileDownloadUri": "http://localhost:8080/downloadFile/Test1.txt",
        "fileType": "text/plain",
        "size": 70
    },
    {
        "fileName": "Test2.txt",
        "fileDownloadUri": "http://localhost:8080/downloadFile/Test2.txt",
        "fileType": "text/plain",
        "size": 70
    }
]
```

➔ **Check File Uploaded or not**

◆ To check file uploaded or not we can use **fileDownloadUri** in the response body

◆ Create a Get request **DownloadSingleFile**

- **GET:** http://localhost:8080/downloadFile/Test1.txt

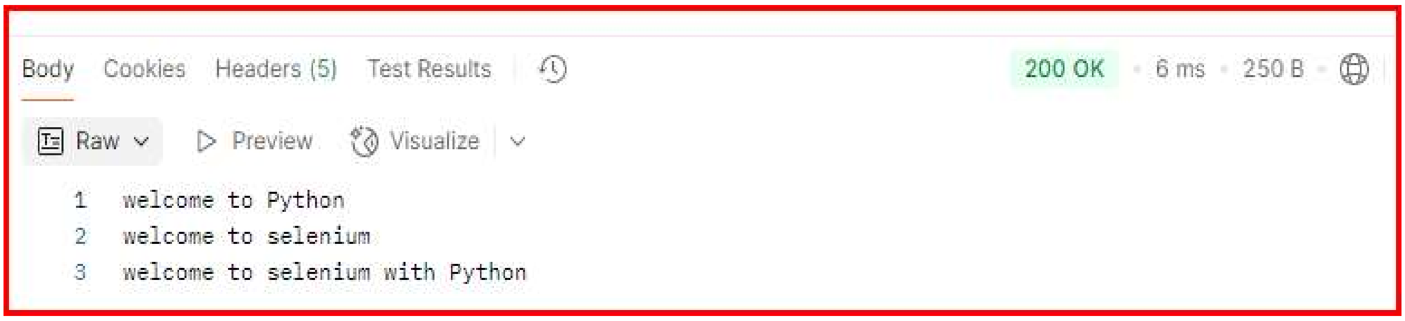

Day7_FileUpload / DownloadSingleFile

GET http://localhost:8080/downloadFile/Test1.txt

Params | Authorization | Headers (6) | Body | Scripts | Settings

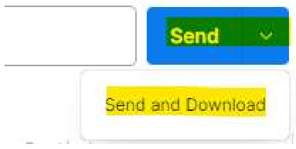
Query Params

Key	Value	Description
Key	Value	Description



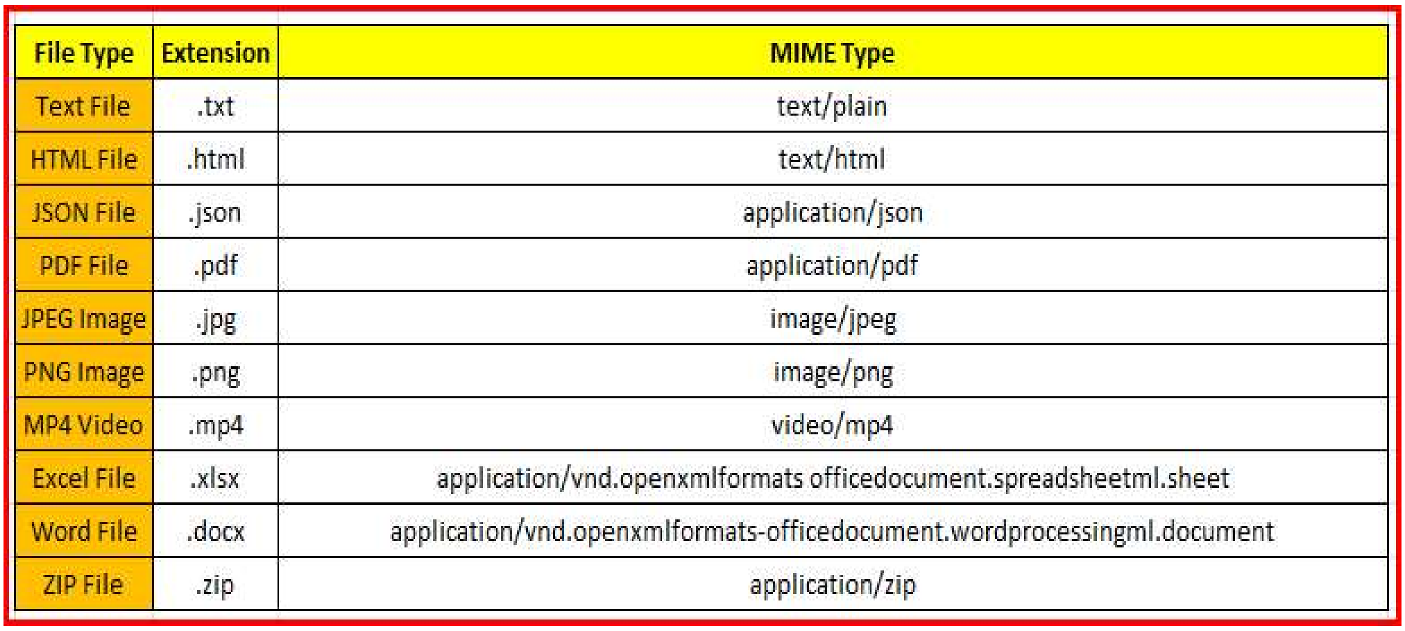
```
welcome to Python
welcome to selenium
welcome to selenium with Python
```

- ➔ It will display contents of file in the Response body. But If we use fileDownloadUri in UI it will download the file.
- ➔ If we want to download the file
  - ◆ **Expand Send** → click on **Send and Download**



Note

- ➔ Every file type has a specific MIME type. MIME type (Multipurpose Internet Mail Extensions) tells the browser or server what kind of file is being sent.



File Type	Extension	MIME Type
Text File	.txt	text/plain
HTML File	.html	text/html
JSON File	.json	application/json
PDF File	.pdf	application/pdf
JPEG Image	.jpg	image/jpeg
PNG Image	.png	image/png
MP4 Video	.mp4	video/mp4
Excel File	.xlsx	application/vnd.openxmlformats officedocument.spreadsheetml.sheet
Word File	.docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document
ZIP File	.zip	application/zip

- ➔ In the CDrive **uploads folder** will be created automatically as specified by default path in jar file which will have files uploaded.
- ➔ Any type of files we can upload accordingly how an api got designed.
- ➔ We can also add some validations in Post-Response Script like **fileName, size, fileType, etc**

Note

- ➔ 201 Created means the server created a new resource (like a new file or record) as a result of your POST request.
- ➔ 200 OK means the request was successful, but the server might not have created anything new — maybe it just accepted or processed your data.
- ➔ In file uploads, some servers respond with 200 if they just received and processed the file without creating a new resource with a unique URL. Others use 201 if they treat the upload as creating a

new resource you can access later.

➔ So both can be correct — it depends on how the server handles the upload!

➔ If a **file upload or download fails**, the server can respond with different error status codes depending on the problem. Here are common ones

Status Code	Meaning	Why It Happens (For File Upload)
400 Bad Request	The request is malformed or missing data	File too large, missing file, or bad format
401 Unauthorized	Not logged in or no permission	User not authorized to upload
403 Forbidden	Permission denied	User doesn't have upload rights
404 Not Found	Upload URL not found	Wrong upload URL or endpoint
413 Payload Too Large	File size exceeds server limit	File is bigger than allowed size
415 Unsupported Media Type	File type not supported	File format not accepted by server
500 Internal Server Error	Server problem	Server crashed or unexpected error
503 Service Unavailable	Server temporarily down	Server overloaded or down

Lab Assignment

➔ Navigate to https://the-internet.herokuapp.com/upload
   - **Upload** a file via postman

➔ Navigate to https://the-internet.herokuapp.com/download
   - **Download** a file via postman