**File Service API with Server and Client**

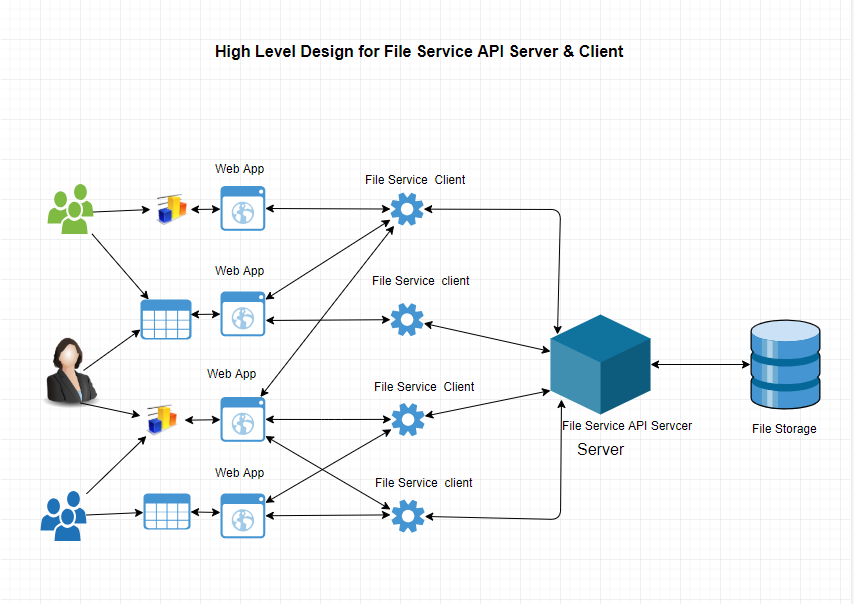
**TASK 1: REST API DESIGN**

Summary:

As you can see in the following diagram 1.1, there is one server application, which takes all Http request from all distributed clients applications to store, update, remove the files, which will have any type like excel, text.

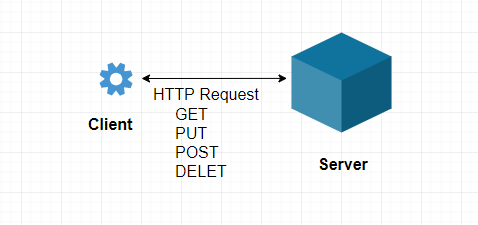
This File Service API will be used by other applications to access the File Service functionality.

High Level Architectural Design of end-to-end system.



Server-Client:

Client will use Http Request to create (HTTP.POST), update (HTTP.PUT), and delete (HTTP.DELETE) and to get files (HTTP.GET). It will use File Service API (Server). You can see in the following diagram.



**TASK 2: Implementation of File Service API( Server Application ) & client Application that will use this File Service.**

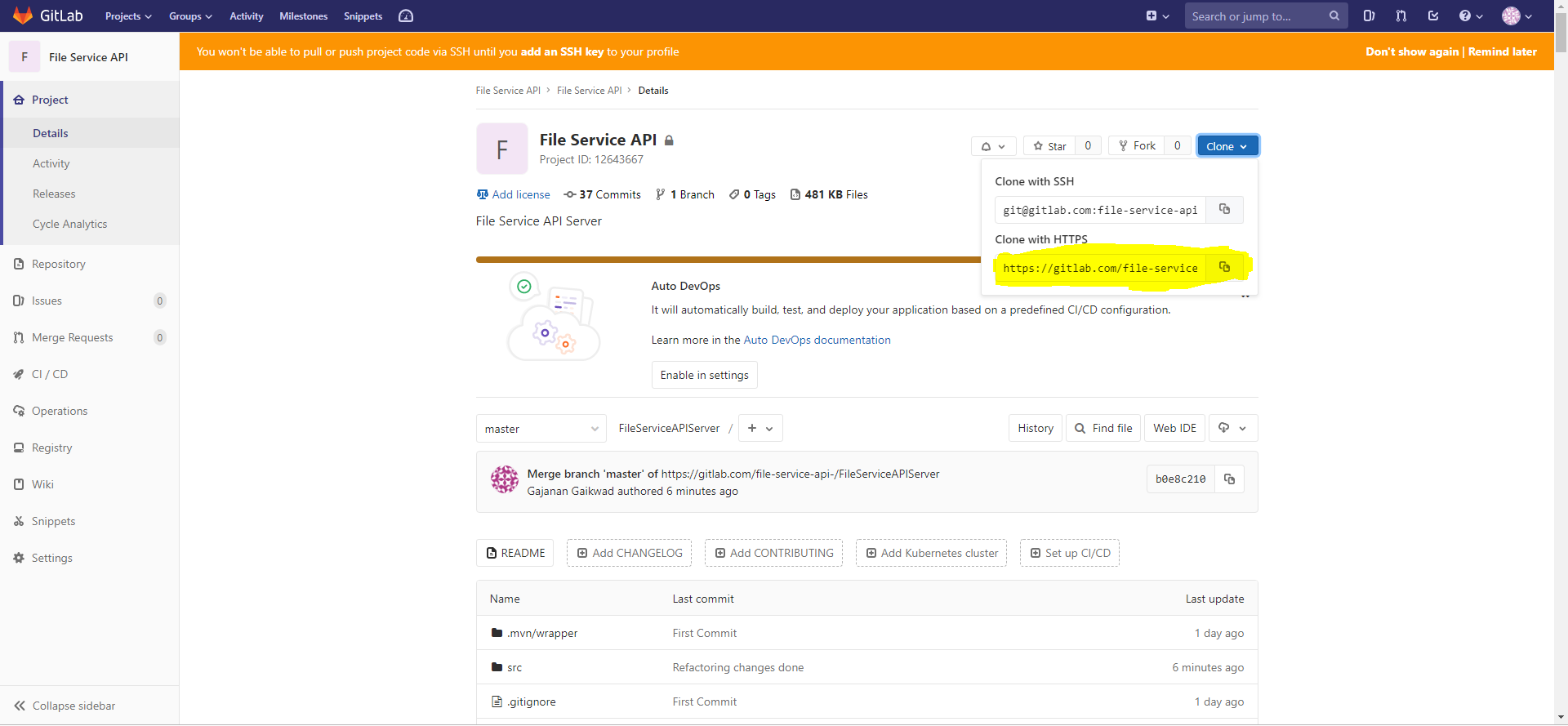
**Installation of File Service API Server**

Step 1: Download the project from gitlab

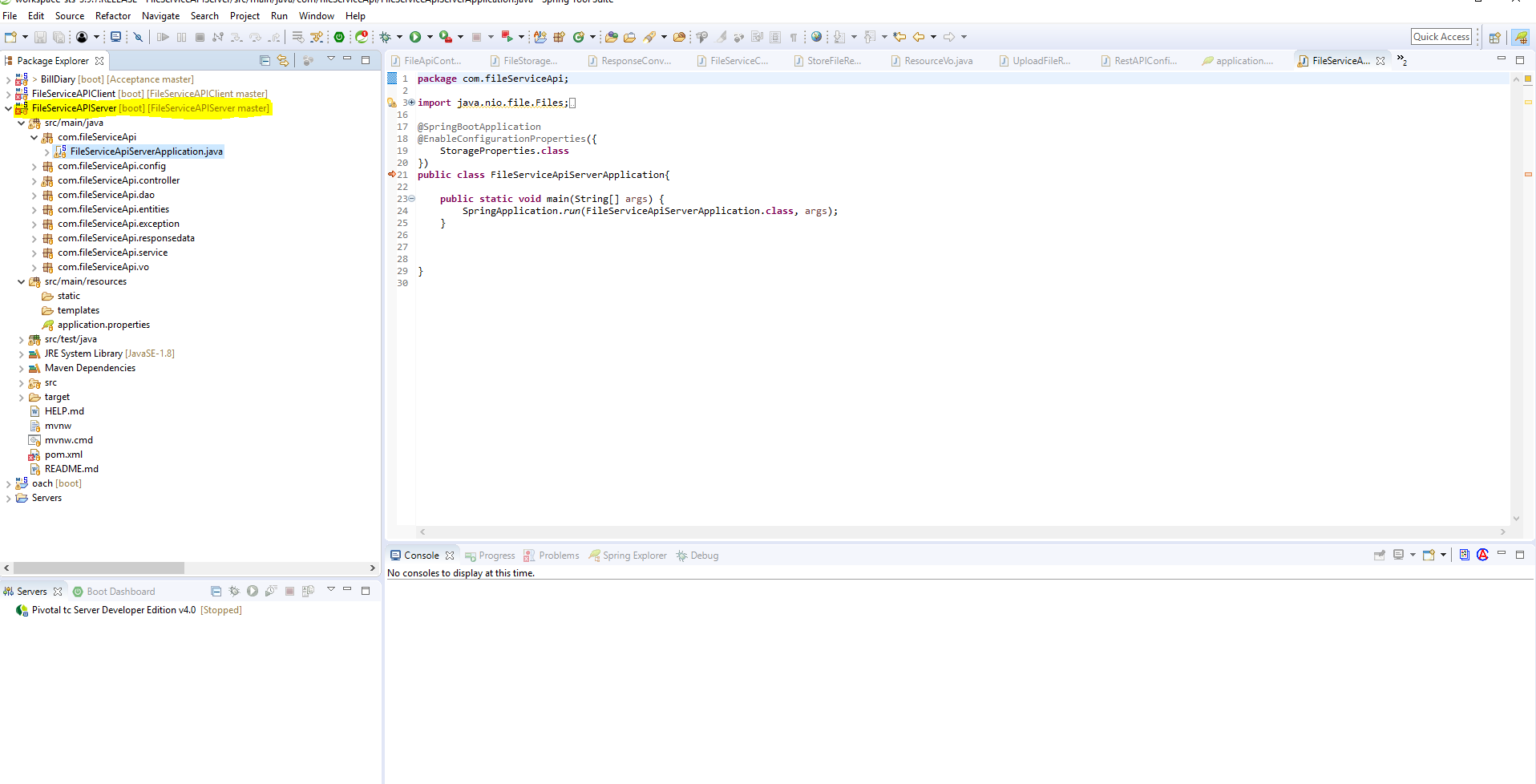
Gitlab URL :[**https://gitlab.com/file-service-api-/FileServiceAPIServer.git**](https://gitlab.com/file-service-api-/FileServiceAPIServer.git)

Use Git Bash to clone the project from the gitlab

**$ git clone https://gitlab.com/file-service-api-/FileServiceAPIServer.git**



Step 2: Open STS, and import the project as maven project



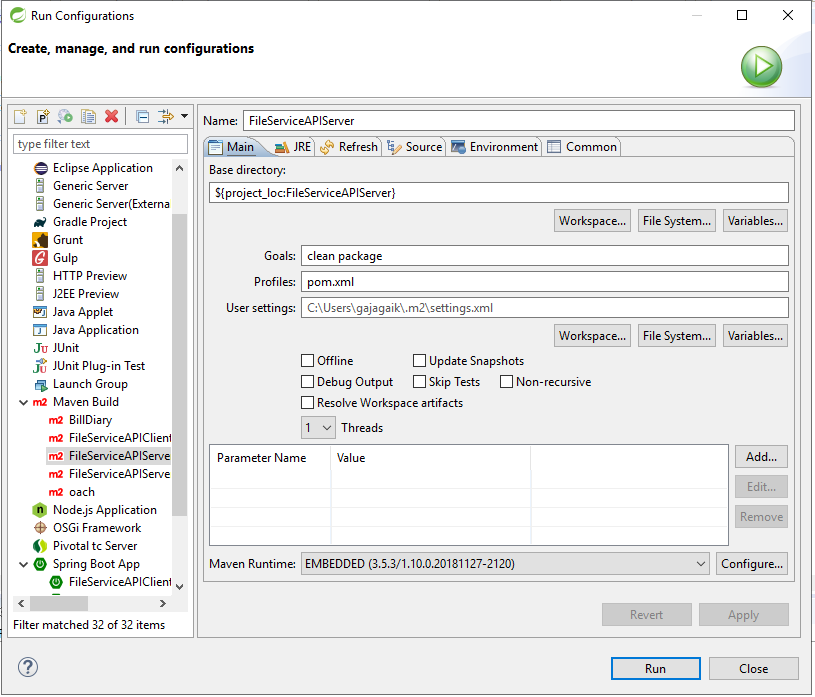
Before running the maven build

Just create a folder structure

C:\Users\gajagaik\FileStorage

And past image file and rename it to cute.jpeg , cause we have written one unit test case so it will execute so for successful completion of the build the TEST case should be passed.

Right click on the project, and Run- > maven-build



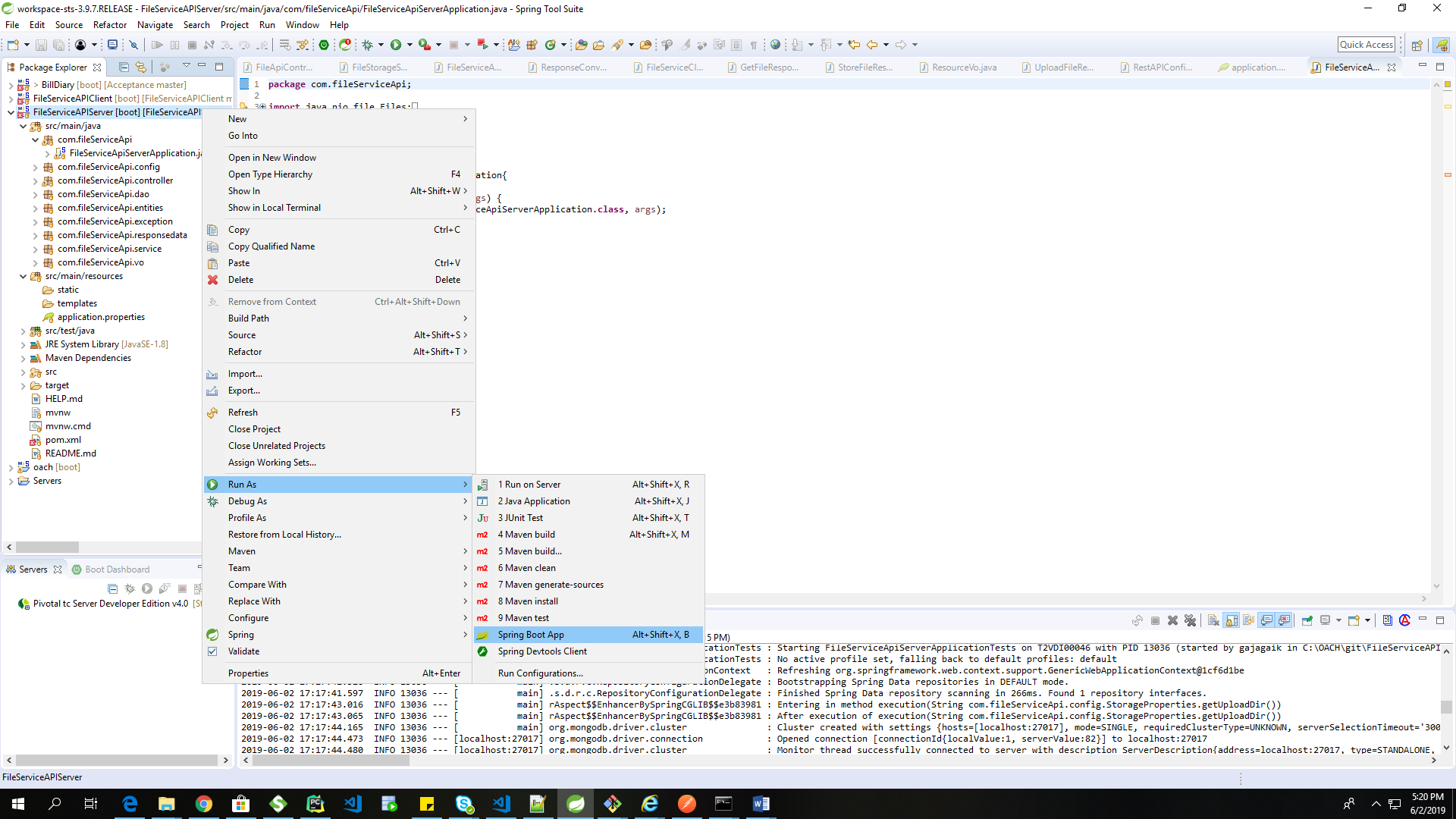
Put the Goal As clean package

And run the project after it will successfully download all the dependency and it will give you the status of the build : SUCCESS

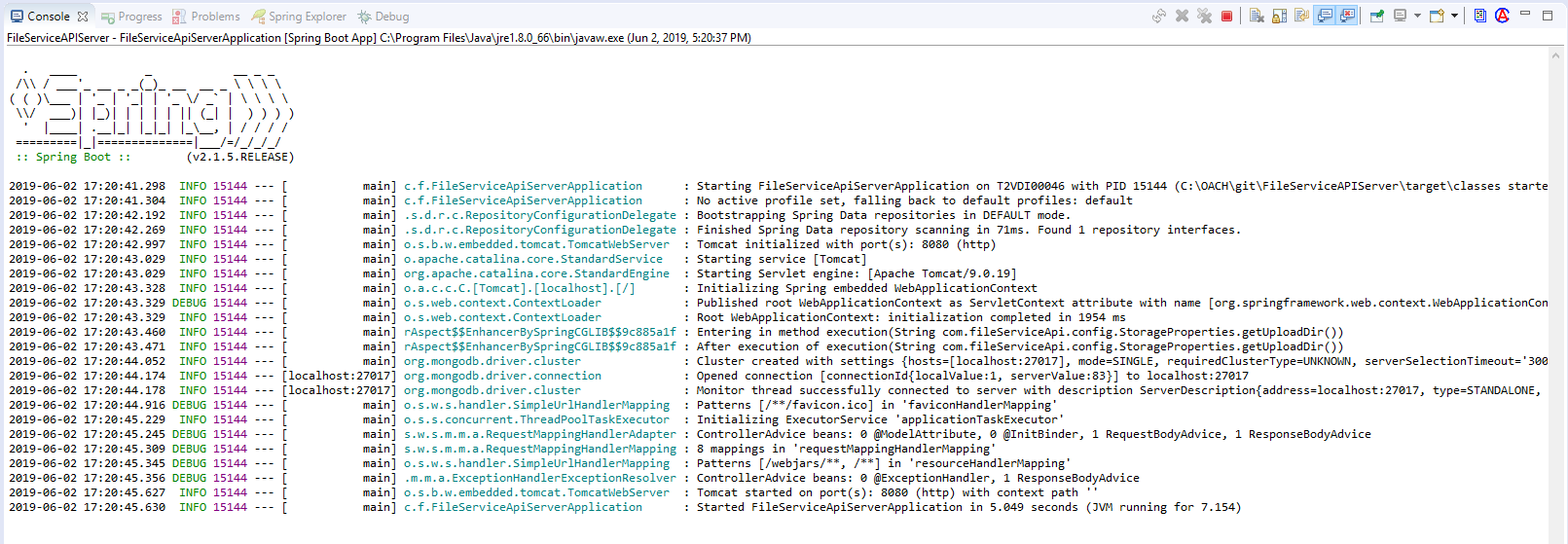


Step 3: Run this project as spring boot application

There are two option either you can run the JAR file or you can run the project by rightclick on the project then run -> Spring Boot Application



After that Server will be started

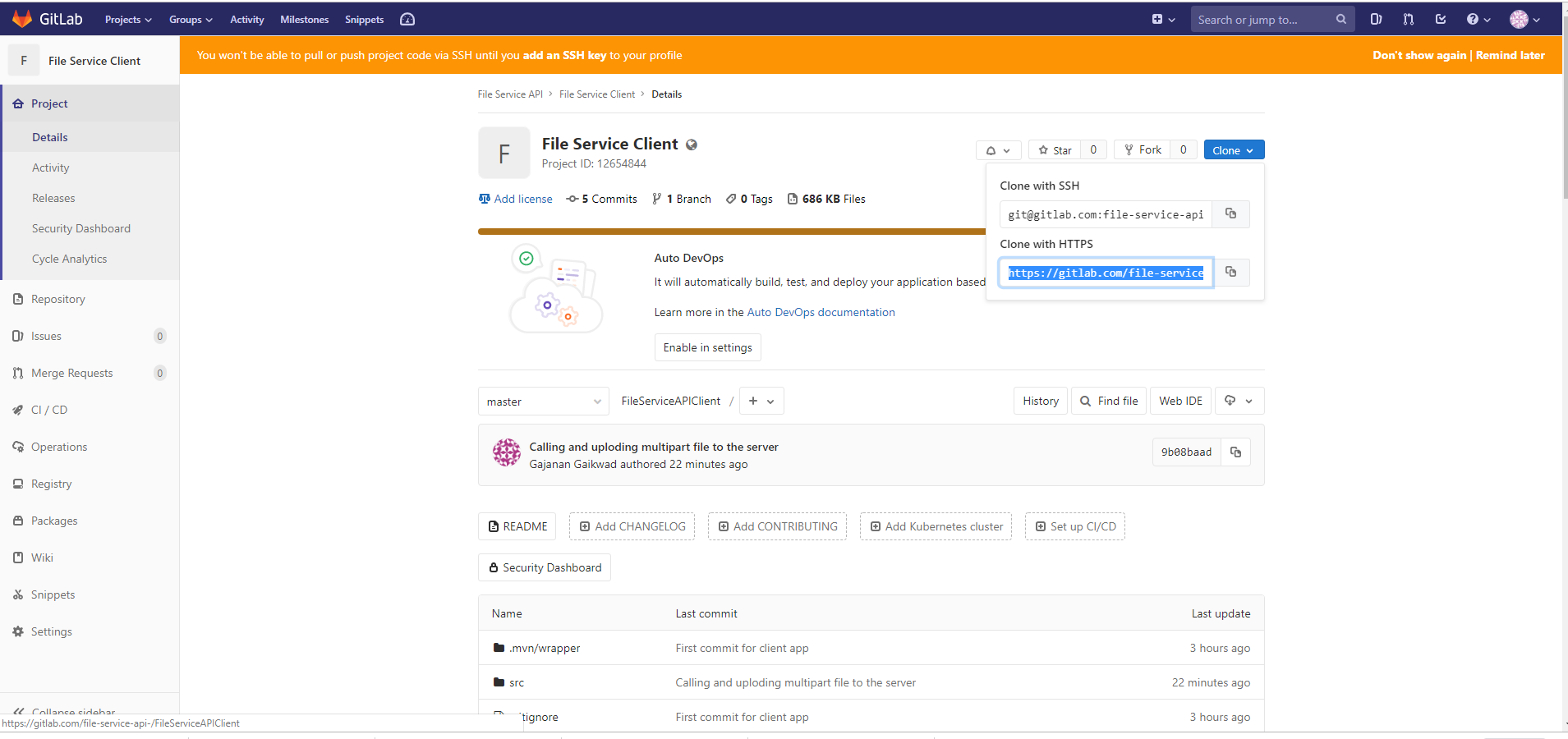


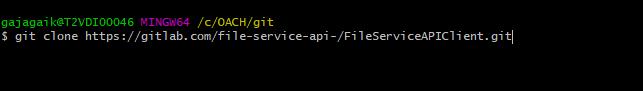
**Installation of File Service API Client**

Now we have to start the client Application

For that also download the project from gitlab using gitbash

GIT: **https://gitlab.com/file-service-api-/FileServiceAPIClient.git**

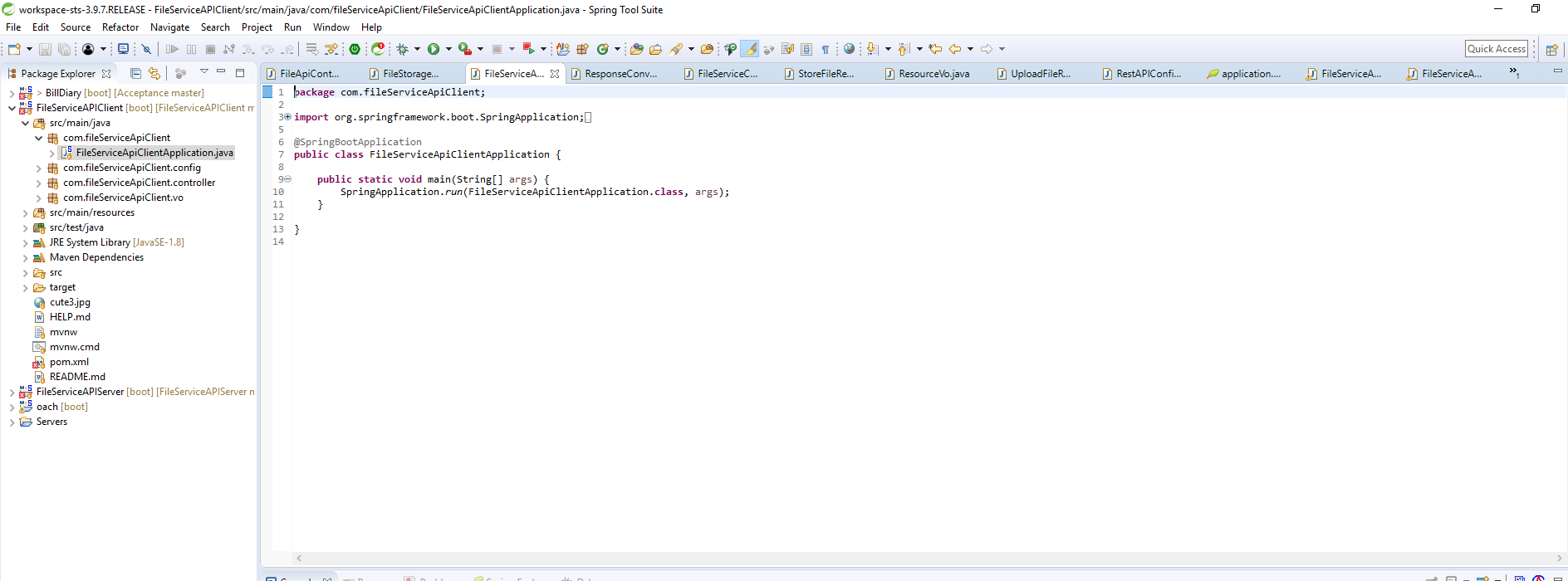




After cloning the project

Open this project in Spring tool Suite (STS)

Same as Server API like Import project as Maven project



After this step Run the maven build for this project also

Goal : clean package

And the run the maven build by using right click on the project -> run -> maven -build



After successful execution it will give message as a

[INFO] BUILD SUCCESS

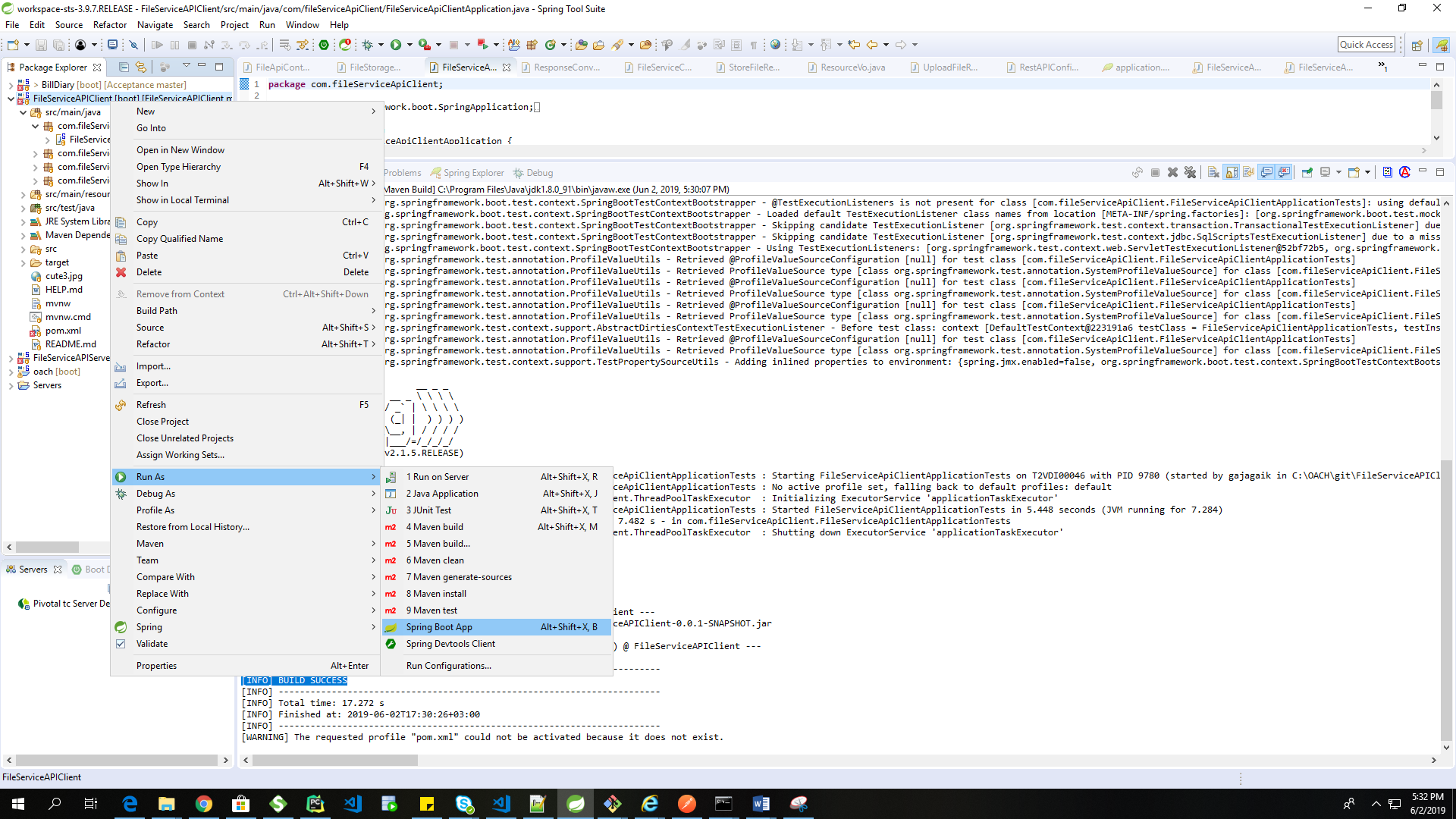
After That,

Please Run the project from taking jar file from /target/ folder

Or

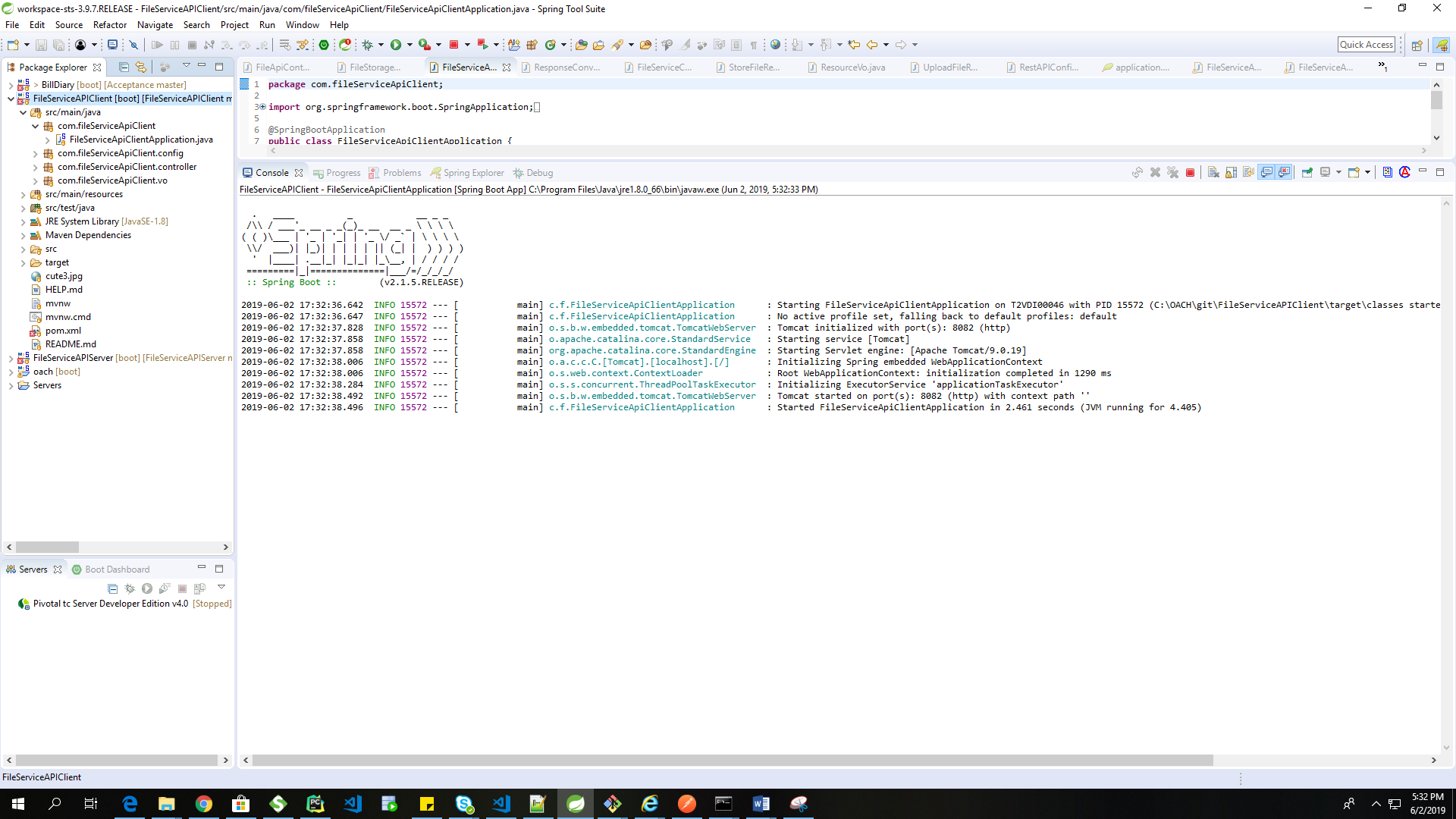
Run the project by right click on it then

Run->Spring Boot Application



Client application will be running on the

http://localhost:8082



So Both Client and Server application will be Running the Successfully

So Now Lets Test These application

**TASK 3**

1. **Unit Testing of File Service API Server:**

We will need a Post Man For That

If you don’t have post man please download it from following URL

<https://www.getpostman.com/downloads/>

after successful installation of postman

open the GUI of POSTMAN application

For All REST Request there is Basic Authentication

Username : admin

Password : password

For Server Testing We will use following REST API URLS

|  |  |  |
| --- | --- | --- |
| Operation | HTTP Method | URL |
| UPLOAD a FILE | POST | localhost:8080/api/upload/ |
| GET the File | GET | localhost:8080/api/download/cute3.jpg |
| Delete the File | DELETE | localhost:8080/api/delete/cute.jpeg |
| Update the File | PUT | localhost:8080/api/update |

For GET & Delete Operation the cute3.jpg and cute.jpeg these are the file names that you can give anything as per your file upload operation

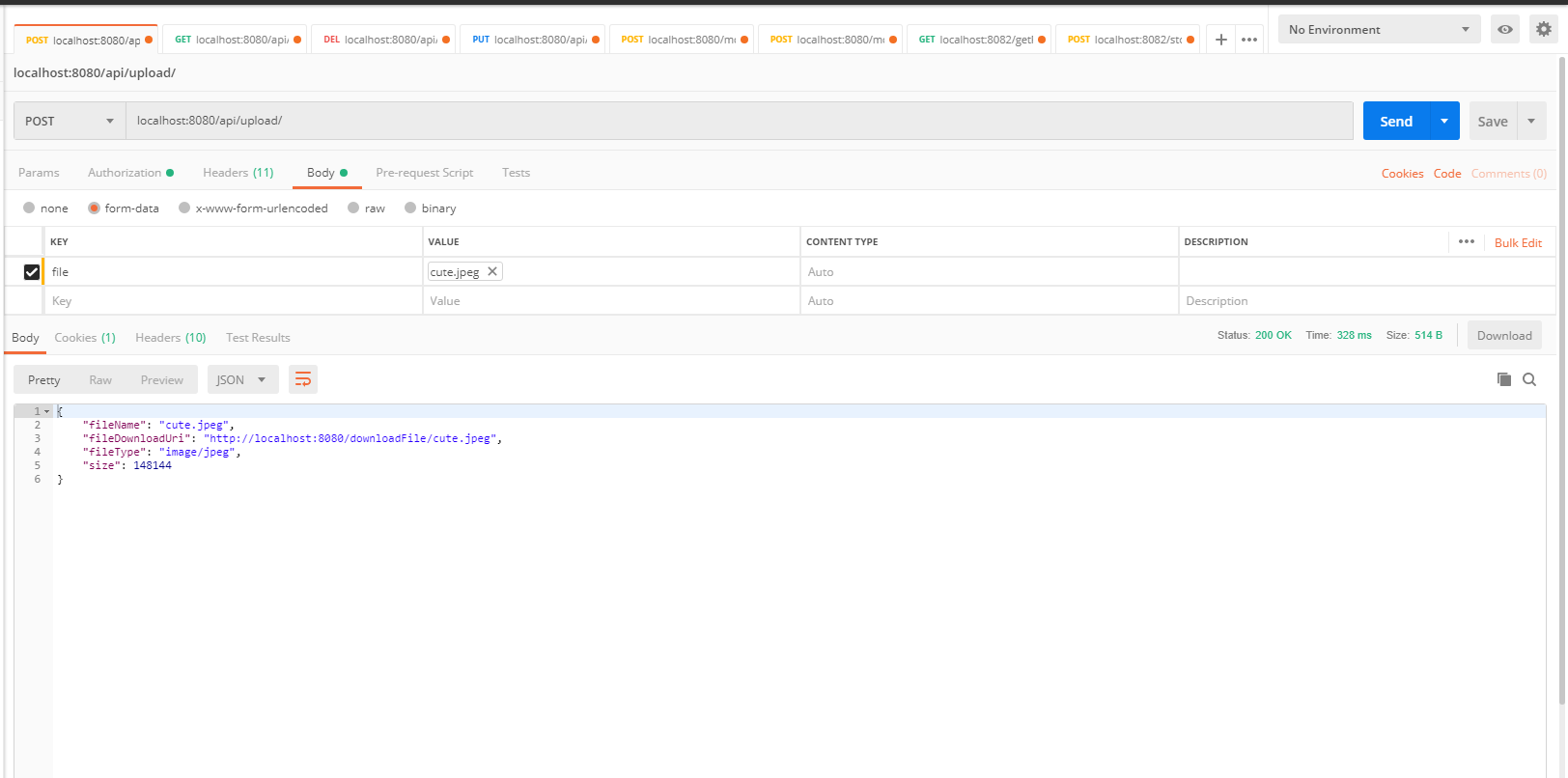
And for PUT method you have to give two rest parameter

1. File
2. File name which we want to update

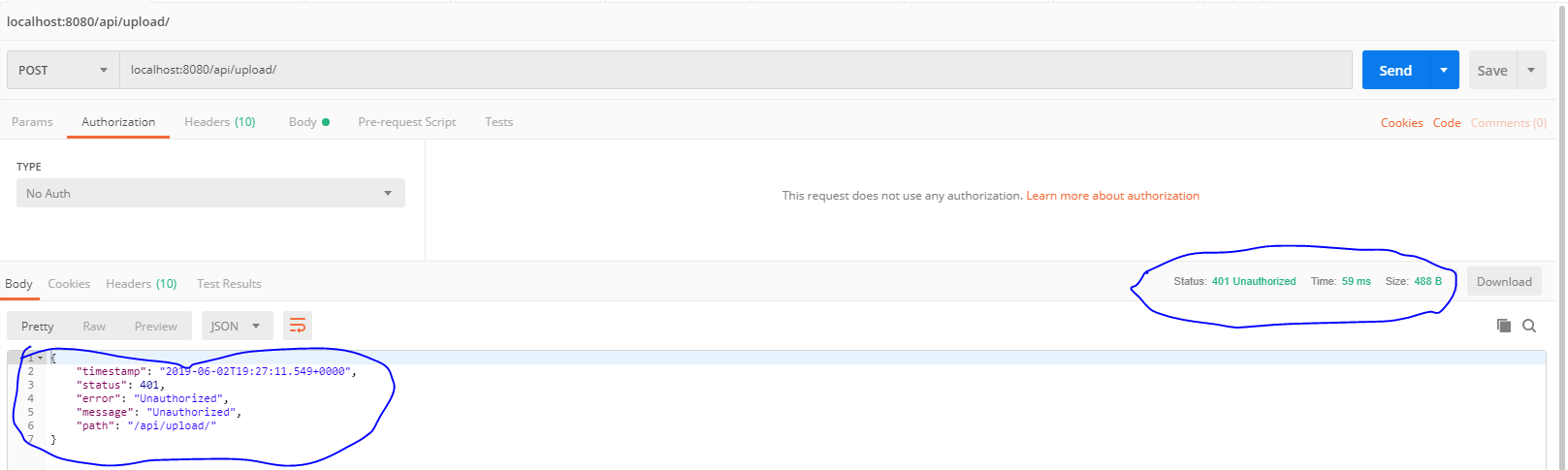
Lets Try

POST Request for File Upload

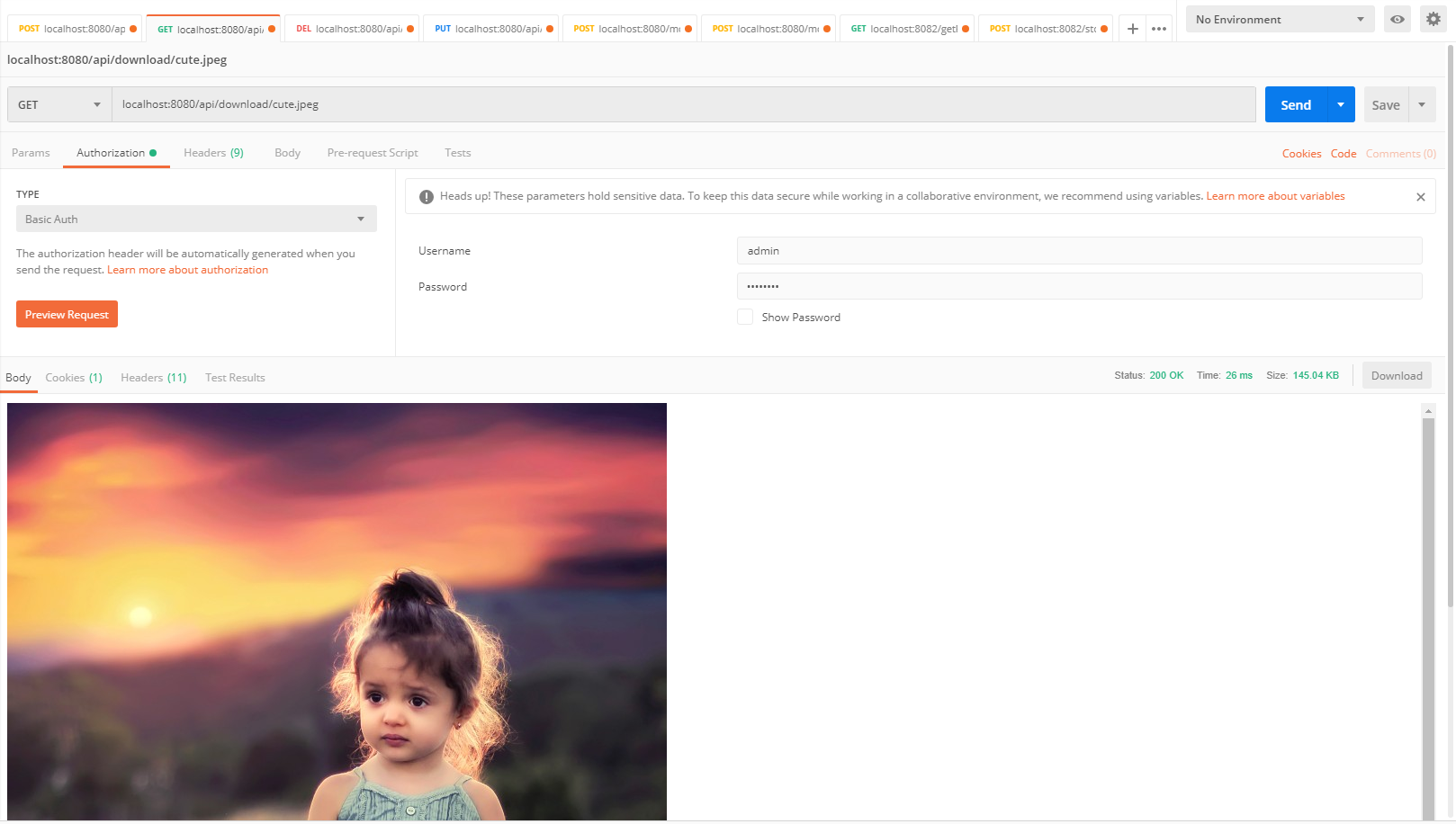
After successful REST call you will get following json response so that file is uploaded we can see, Don’t forget to add authentication i.e username and password



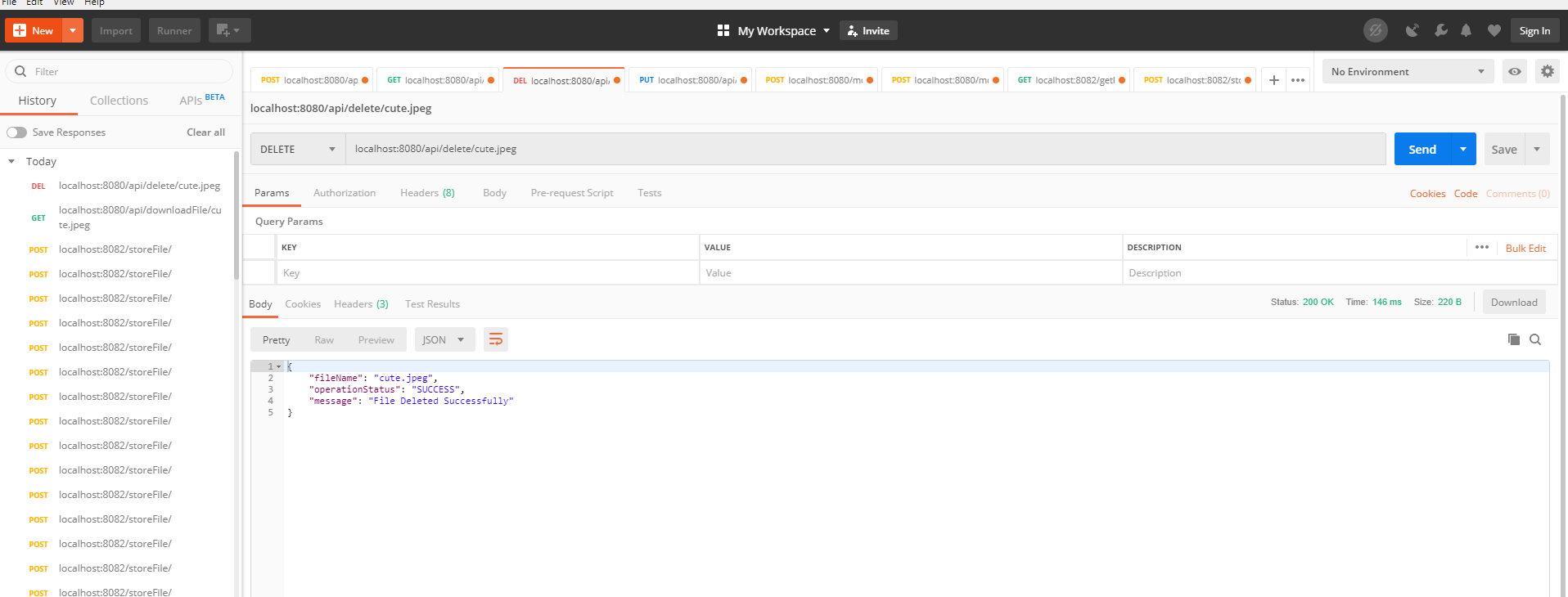
If you don’t user username and password you will get following response



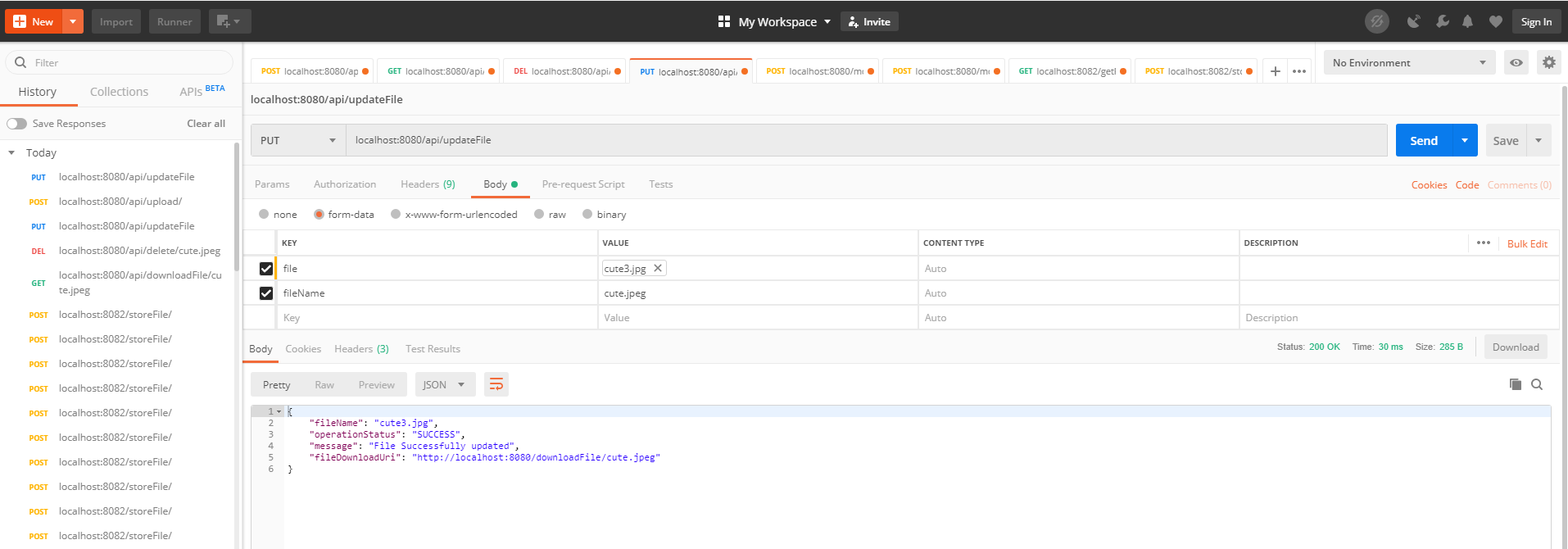
GET Request for File fetch from the File System



Delete Request to delete the file from the File System



PUT Request to update the file in the data center



1. **End To End Testing**

For our client we have the following URLS to send, update, delete and fetch the file from the server. **Note Out Client application is running on 8082 port.**

|  |  |  |
| --- | --- | --- |
| Operation | Method | URL |
| Fetch the File | GET | localhost:8082/getFile/cute3.jpg |
| Store the File | POST | localhost:8082/storeFile/ |
|  |  |  |

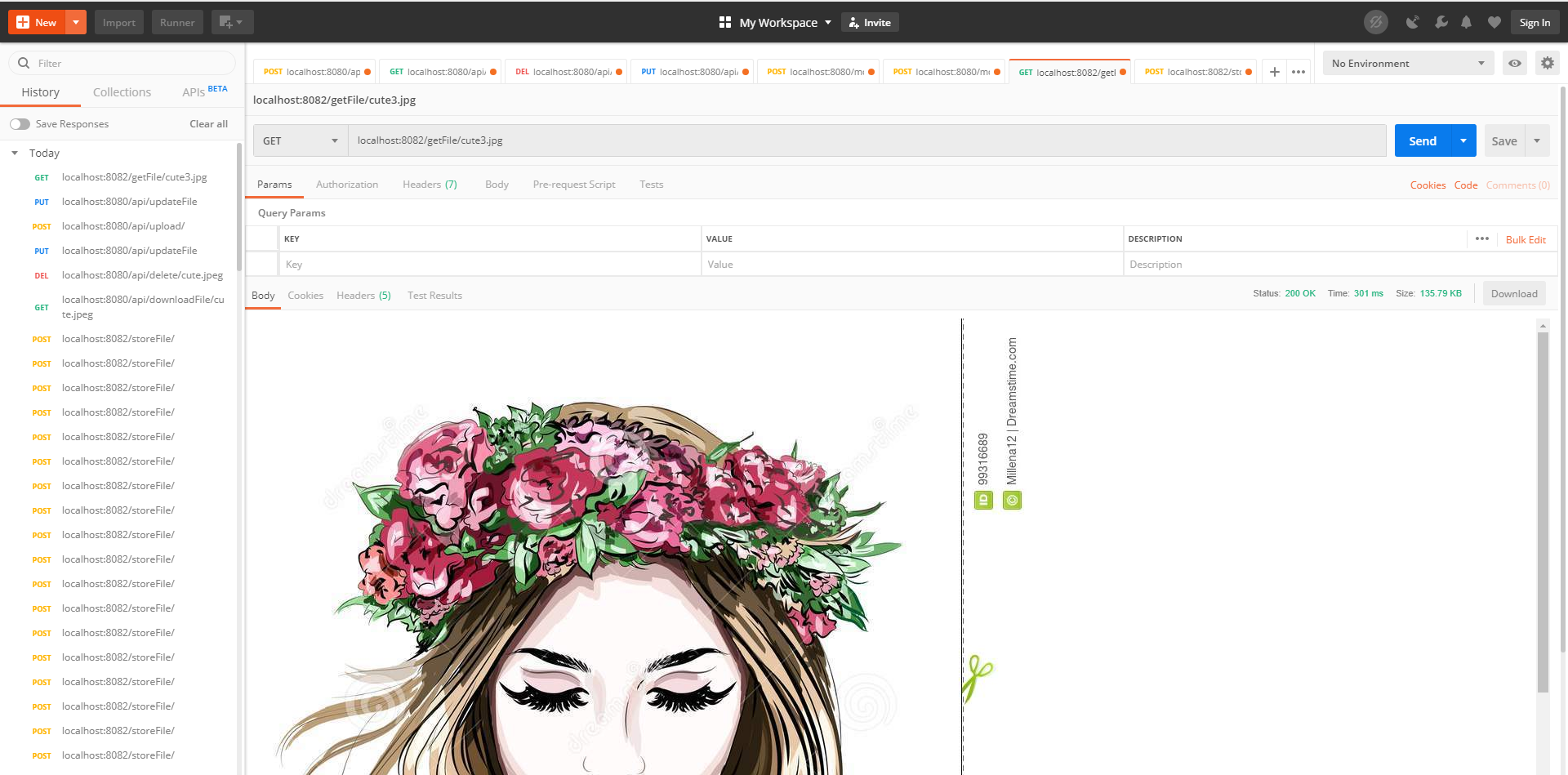
Lets Try

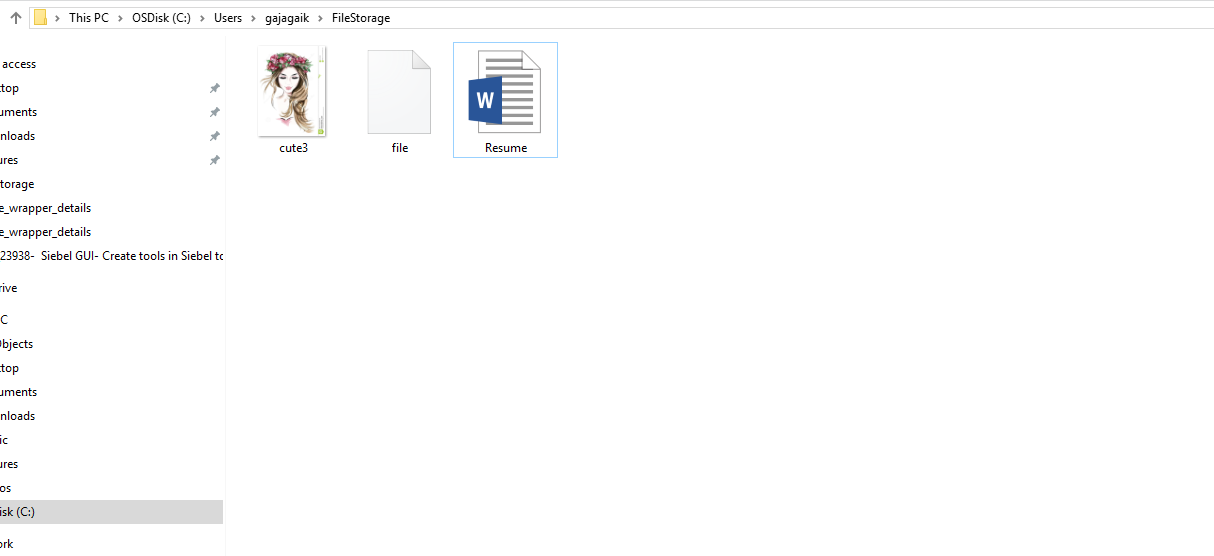
For GET Operation

URL : **localhost:8082/getFile/cute3.jpg**

From the below diagram you can see we are tying the fetch the file through our client API and

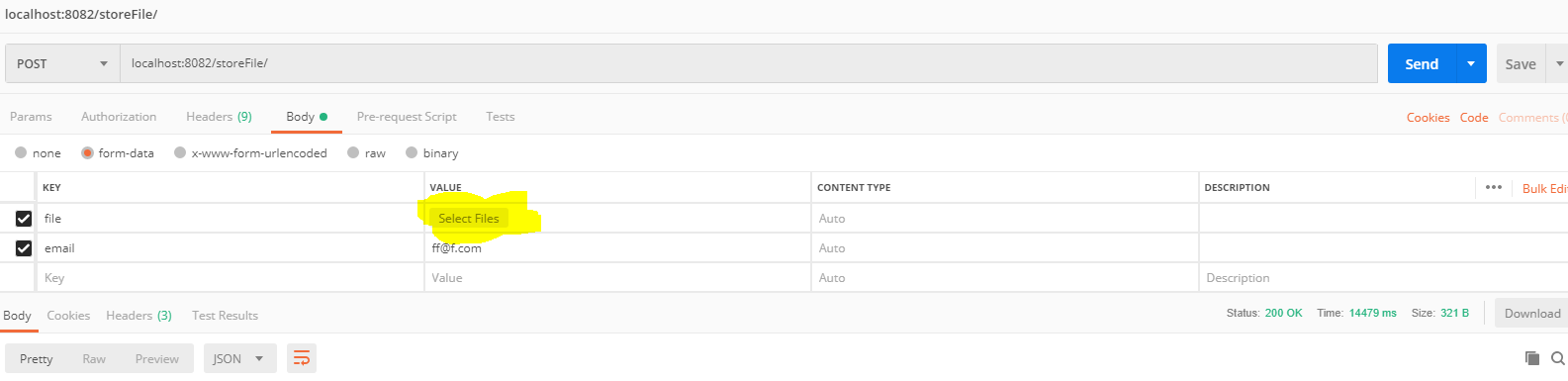
We are fetching file from server data center





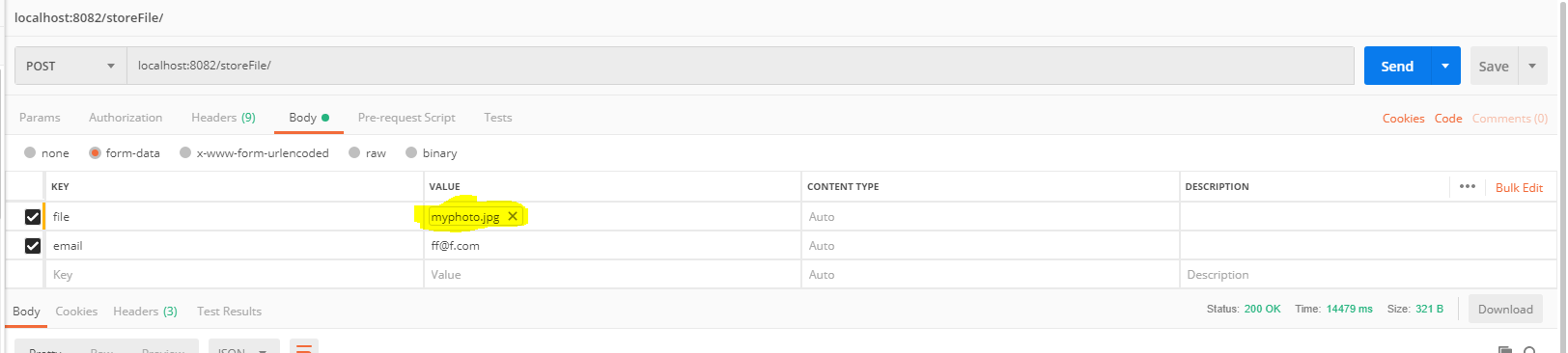
Now Lets try Store operation

In below screen shot you can see there is option for you you can select the file from your local system so click on that and select the file

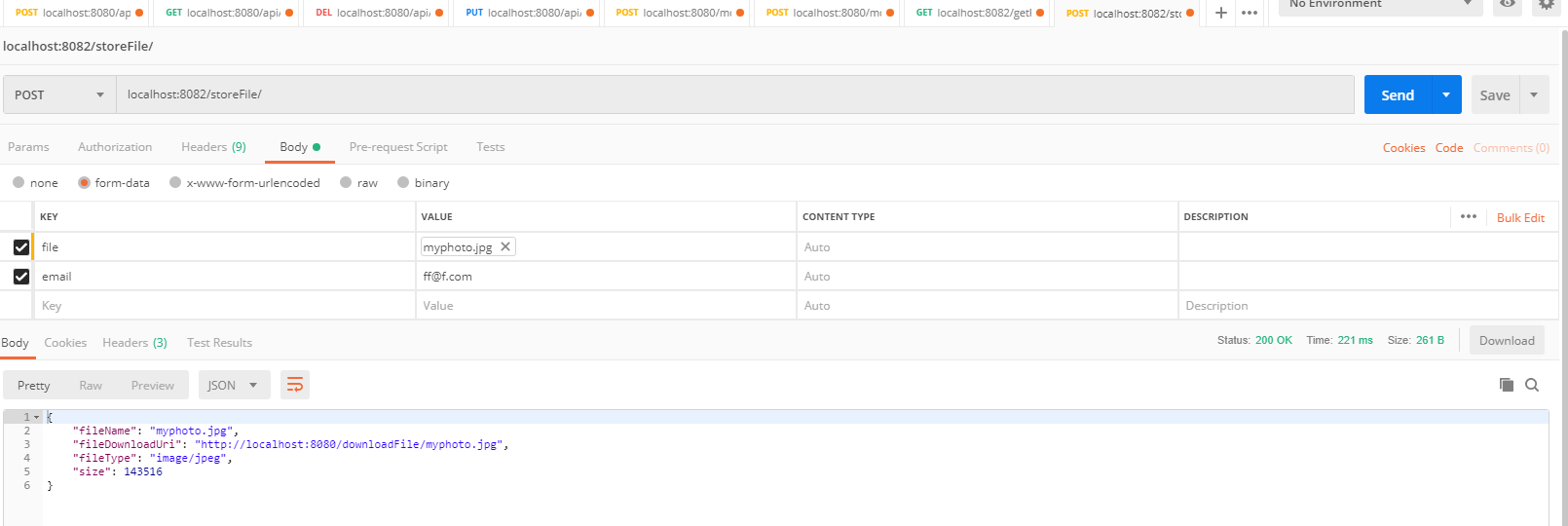


Following screenshot you can see that I have selected myPhoto.jpg file

And let hit the webservice

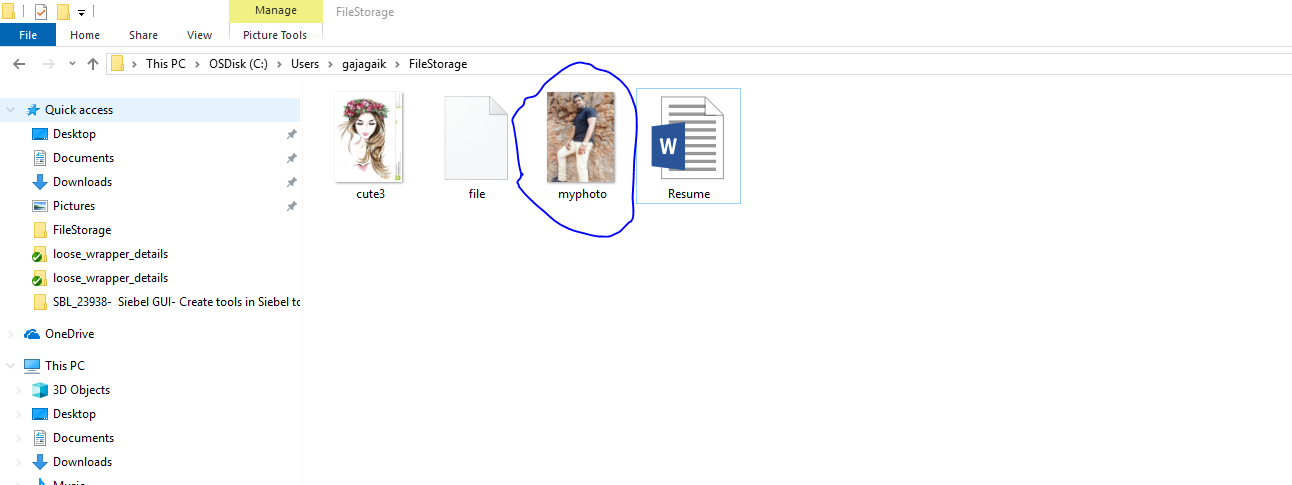


We will get Following Response that means our file has been stored successfully



Lets Check the file in local storage,

**Ohhhh ! Wow our file has been stored successfully**



**So End to End Test Successfully passed.**