# README:

## Clone the Project:

Use the below GitHub URL to clone the project (**employee-inventory**)

https://github.com/pd100rab/pd100rab

## Download Dependencies

Use *mvn clean install* command to download all the required dependencies.

## Run the project

Bootstrap the Spring Boot project by running the main Application.java

OR

Run Spring Boot application with command: mvn spring-boot:run

## API Testing

In order to test the below APIs, please use any API Testing tool like Postman.

List of API:

### uploadEmployeeInfo – POST request to upload a File

This API is used to upload a File into a Directory. Once uploaded, the API will read the same file and process the data. Once processing gets completed, the data gets stored into the Database based on status (COMPLETED, COMPLETED\_WITH\_ERRORS, FAILED).

The API response will provide further useful information as name, status API URL, status, number of success data, number of failed data.

URL - http://localhost:8080/api/employee

Body - provide “file” as key and select the particular file as value.

### getStatus – GET request to fetch status of the process

This API will return the status information of the process based on the File Id provided in the request.

URL - http://localhost:8080/api/status/ffe88d55-d504-4c2c-96f8-068b2d571066

### updateEmployee – PUT request to update employee information

This API is used to update any employee information.

URL - <http://localhost:8080/api/employee/3>

Body – Provide the Employee object as JSON

Eg - {

"id": 3,

"fName": "RAHUL",

"lName": "DRAVID",

"age": 33

}

### fetchAllEmployees – GET request to fetch all employees details

Return the list of employees stored in the DB.

URL - http://localhost:8080/api/fetchAllEmployees

### deleteEmployee – DELETE request to delete any particular employee

Returns 204 No Content after deleting the record based on id.

URL - http://localhost:8080/api/employee/2

### deleteAllEmployees – DELETE request to delete all employees

Returns 204 No Content after deleting all the records.

URL - http://localhost:8080/api/employee

## Database

As database, I have used H2 DB which is an In Memory Database provided by Spring Boot.

Console URL - http://localhost:8080/h2-console/

The 2 Tables namely EMPLOYEE and FILE\_DB can be seen created as the server gets started.

## JUnit

I tried to cover JUnit test cases of all the important classes.

## Future Scopes

1. Logging Mechanism can be added
2. Security layer can be added
3. Deployment to any cloud infrastructure can be done