**Create a project to upload and download a file in Spring Boot.**

Pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.7.17</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>coms</groupId>

<artifactId>FileUploadAndDownloadServiceApp</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>FileUploadAndDownloadServiceApp</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-tomcat</artifactId>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

Application.properties

server.port = 9097

#Db Configurations

spring.datasource.url=jdbc:mysql://localhost:3306/phase3

spring.datasource.username=root

spring.datasource.password=sonikaSHETTY#17

#H2 Db Configurations

#spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

#spring.datasource.url=jdbc:h2:mem:testdb

#spring.datasource.driverClassName=org.h2.Driver

#spring.datasource.username=sa

#spring.datasource.password=

#spring.h2.console.enabled=true

spring.jpa.hibernate.ddl-auto=update

java file

package coms.FileUploadAndDownloadServiceApp;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@SpringBootApplication*

public class FileUploadAndDownloadServiceAppApplication {

public static void main(String[] args) {

SpringApplication.*run*(FileUploadAndDownloadServiceAppApplication.class, args);

System.***out***.println("Service Started...");

}

}

package coms.FileUploadAndDownloadServiceApp.controllers;

import java.util.ArrayList;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import org.springframework.web.multipart.MultipartFile;

import org.springframework.web.servlet.support.ServletUriComponentsBuilder;

import coms.FileUploadAndDownloadServiceApp.model.FileInfo;

import coms.FileUploadAndDownloadServiceApp.model.ResponseUrl;

import coms.FileUploadAndDownloadServiceApp.service.FileInfoService;

import org.springframework.http.HttpHeaders;

*@RestController*

*@RequestMapping*("/api/v1")

public class FileInfoController {

*@Autowired*

FileInfoService fis;

*@PostMapping*("/fupload")

public ResponseEntity<Object> FileSaving(*@RequestParam*("file") MultipartFile file)

{

//System.out.println("Hello");

if(fis.FileUpload(file).equals("Success"))

return new ResponseEntity<Object>("File Uploaded Successfully", *HttpStatus*.***CREATED***);

return new ResponseEntity<Object>("Error ! File Not Uploaded Successfully", *HttpStatus*.***OK***);

}

*@GetMapping*("/showall")

public ResponseEntity<List<ResponseUrl>> GetAllInfo()

{

List<ResponseUrl> responseAll = new ArrayList<ResponseUrl>();

ResponseUrl rinfo = null;

List<FileInfo> fall = fis.ShowAll();

// Generating URL for Download

for(FileInfo f : fall)

{

String fileDownloadUrl = ServletUriComponentsBuilder

.*fromCurrentContextPath*() // servername : http://localhost:9092

.path("api/v1/files/") // getmapping text

.path(f.getFileid()) // id value from db

.toUriString();

System.***out***.println(fileDownloadUrl);

rinfo = new ResponseUrl();

rinfo.setFilename(f.getFilename());

rinfo.setFilesize(f.getFiledata().length);

rinfo.setFileUrl(fileDownloadUrl);

responseAll.add(rinfo);

}

return new ResponseEntity<List<ResponseUrl>>(responseAll, *HttpStatus*.***OK***);

}

*@GetMapping*("/files/{id}")

public ResponseEntity<byte[]> GetFileDownload(*@PathVariable*("id") String id)

{

FileInfo finfo = fis.GetAFile(id);

if(finfo!=null)

{

// download the file

return ResponseEntity.*ok*()

.header(HttpHeaders.***CONTENT\_DISPOSITION***, "attachment; filename=\"" + finfo.getFilename() + "\"")

.body(finfo.getFiledata());

}

return new ResponseEntity("File Not Found", *HttpStatus*.***NOT\_FOUND***);

}

}

package coms.FileUploadAndDownloadServiceApp.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.Lob;

import javax.persistence.Table;

import org.hibernate.annotations.GenericGenerator;

*@Entity*

*@Table*(name="FileInfo\_RestApi")

public class FileInfo {

*@Id*

*@GeneratedValue*(generator = "uuid")

*@GenericGenerator*(name = "uuid", strategy = "uuid2")

private String fileid;

private String filename;

private String filetype;

*@Lob* // it will refer in database column type as Blob(Binary Large Objects).

private byte[] filedata;

public String getFileid() {

return fileid;

}

public void setFileid(String fileid) {

this.fileid = fileid;

}

public String getFilename() {

return filename;

}

public void setFilename(String filename) {

this.filename = filename;

}

public String getFiletype() {

return filetype;

}

public void setFiletype(String filetype) {

this.filetype = filetype;

}

public byte[] getFiledata() {

return filedata;

}

public void setFiledata(byte[] filedata) {

this.filedata = filedata;

}

}

package coms.FileUploadAndDownloadServiceApp.model;

public class ResponseUrl {

private String filename;

private int filesize;

private String fileUrl;

public String getFilename() {

return filename;

}

public void setFilename(String filename) {

this.filename = filename;

}

public int getFilesize() {

return filesize;

}

public void setFilesize(int filesize) {

this.filesize = filesize;

}

public String getFileUrl() {

return fileUrl;

}

public void setFileUrl(String fileUrl) {

this.fileUrl = fileUrl;

}

}

package coms.FileUploadAndDownloadServiceApp.repo;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import coms.FileUploadAndDownloadServiceApp.model.FileInfo;

*@Repository*

public interface FileInfoRepository extends JpaRepository<FileInfo, String> {

}

package coms.FileUploadAndDownloadServiceApp.service;

import java.util.List;

import org.springframework.web.multipart.MultipartFile;

import coms.FileUploadAndDownloadServiceApp.model.FileInfo;

public interface FileInfoService {

public String FileUpload(MultipartFile file);

public List<FileInfo> ShowAll();

public FileInfo GetAFile(String fileid);

}

package coms.FileUploadAndDownloadServiceApp.service;

import java.io.IOException;

import java.util.List;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.web.multipart.MultipartFile;

import coms.FileUploadAndDownloadServiceApp.model.FileInfo;

import coms.FileUploadAndDownloadServiceApp.repo.FileInfoRepository;

*@Service*

public class FileInfoServiceImpl implements FileInfoService {

*@Autowired*

FileInfoRepository fir;

*@Override*

public String FileUpload(MultipartFile file) {

String res = "err";

try

{

FileInfo fileinfo = new FileInfo();

fileinfo.setFilename(file.getOriginalFilename());

fileinfo.setFiletype(file.getContentType());

fileinfo.setFiledata(file.getBytes());

FileInfo f = fir.save(fileinfo);

if(f!=null)

res ="Success";

} catch (IOException e) {

// **TODO** Auto-generated catch block

System.***out***.println(e);

}

return res;

}

*@Override*

public List<FileInfo> ShowAll() {

// **TODO** Auto-generated method stub

return fir.findAll();

}

*@Override*

public FileInfo GetAFile(String fileid) {

Optional<FileInfo> fs = fir.findById(fileid);

if(fs.isPresent())

return fs.get();

return null;

}

}



