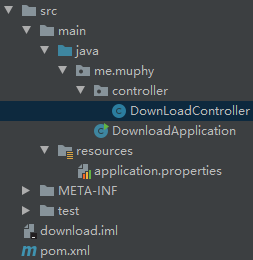
**本地文件下载（上传），可启动服务代替SCP命令共享文件**

**新建spring-boot项目**



**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>2.2.5.RELEASE</version>

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

</parent>

<groupId>me.muphy</groupId>

<artifactId>download</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>download</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-web</artifactId>

</dependency>

<dependency>

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

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

<scope>test</scope>

<exclusions>

<exclusion>

<groupId>org.junit.vintage</groupId>

<artifactId>junit-vintage-engine</artifactId>

</exclusion>

</exclusions>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**application.properties**

server.port=8088

#文件下载

#download.file.default=20200428.xls

download.path=E:/workspace/download/

**DownloadApplication.java**

package me.muphy;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DownloadApplication {

public static void main(String[] args) throws InterruptedException {

SpringApplication.run(DownloadApplication.class, args);

}

}

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>莫非录音机</title>

</head>

<body>

<div>

<div><span><a href="/ll">文件下载</a></span></div>

<div><span><a href="/upload/index.html">文件上传</a></span></div>

<div><span><a href="/record/index.html">录音机</a></span></div>

</div>

</body>

</html>

**DownLoadController.java**

package me.muphy.download;

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

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

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

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.\*;

import java.util.ArrayList;

import java.util.List;

@RestController

public class DownLoadController {

@Value("${download.file.default:}")

private String downloadDefaultFile;

@Value("${download.path:E:/workspace/download/}")

private String downloadPath;

@Value("${download.url:http://(ip):(port)}")

private String downloadUrl;

@GetMapping("/dl")

public void download(String f, HttpServletResponse response) throws IOException {

// 下载本地文件

File baseFile = new File(downloadPath);

String fileName = downloadDefaultFile; // 文件的默认保存名

if (f != null && !"".equals(f.trim())) {

fileName = f;

}

File file = new File(downloadPath + fileName);

if(!file.isFile() || !file.getCanonicalPath().startsWith(baseFile.getCanonicalPath())){

error(response);

return;

}

// 读到流中

InputStream inStream = new FileInputStream(file);// 文件的存放路径

// 设置输出的格式

response.reset();

response.setContentType("bin");

response.addHeader("Content-Disposition", "attachment; filename=\"" + file.getName() + "\"");

// 循环取出流中的数据

byte[] b = new byte[100];

int len;

while ((len = inStream.read(b)) > 0)

response.getOutputStream().write(b, 0, len);

inStream.close();

}

private void error(HttpServletResponse response) throws IOException {

response.reset();

response.setContentType("text/html;charset=utf-8");

PrintWriter writer = new PrintWriter(new OutputStreamWriter(response.getOutputStream(), "utf-8"));

writer.print("文件不存在!");

writer.close();

return;

}

@GetMapping("/ll")

public String ListFiles(String d, HttpServletRequest request) throws IOException {

File baseFile = new File(downloadPath);

if (!baseFile.isDirectory()) {

System.out.println(downloadPath + " 目录不存在，只能下载此目录下面的文件，请确保配置路径（download.path={path}）存在");

return "目录不存在!";

}

String ip = request.getServerName();

int port = request.getServerPort();

List<String> fns = new ArrayList<>();

String path = baseFile.getAbsolutePath();

if(d != null && !d.isEmpty()){

path += d;

}

File file = new File(path.replaceAll("/+", "/"));

if (file.isDirectory()) {

file = file.getCanonicalFile();

File[] files = file.listFiles();

for (int i = 0; i < files.length; i++) {

if (files[i].isDirectory()) {

fns.add("<span>目录：<a href=\"" + downloadUrl.replace("(ip)", ip).replace("(port)", port + "") + "/ll?d=" + getQueryParameter(files[i], baseFile) + "\" >" + files[i].getName() + "</a></span>");

continue;

}

fns.add("<span>文件：<a href=\"" + downloadUrl.replace("(ip)", ip).replace("(port)", port + "") + "/dl?f=" + getQueryParameter(files[i], baseFile) + "\" >" + files[i].getName() + "</a></span>");

}

} else {

return "目录不存在!";

}

fns.sort((x,y)->y.compareTo(x));

if(!file.getCanonicalPath().equals(baseFile.getAbsolutePath())){

fns.add(0, "<span>目录：<a href=\"" + downloadUrl.replace("(ip)", ip) + "/ll?d=" + getQueryParameter(file, baseFile) + "/..\" >..</a></span>");

}

return String.join("<br>", fns);

}

private String getQueryParameter(File file, File baseFile) throws IOException {

return file.getCanonicalPath()

.replaceAll("\\\\", "/")

.replaceAll(baseFile.getAbsolutePath().replaceAll("\\\\", "/"), "" );

}

}

新增文件上传代码

upload.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>文件上传</title>

</head>

<body>

<div>

<h1>文件上传</h1>

<form action="/ul" method="post" enctype="multipart/form-data">

<p>选择文件: <input type="file" name="fileName"/></p>

<p>选择文件: <input type="file" name="fileName"/></p>

<p><input type="submit" value="提交"/></p>

</form>

</div>

</body>

</html>

UploadController.java

package me.muphy.upload;

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

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

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

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

import org.springframework.web.multipart.MultipartFile;

import org.springframework.web.multipart.MultipartHttpServletRequest;

import javax.servlet.http.HttpServletRequest;

import java.io.File;

import java.io.IOException;

import java.util.List;

@RestController

public class UploadController {

@Value("${download.path:E:/workspace/download/}")

private String uploadPath;

/\*\*

\* 实现多文件上传

\*/

@PostMapping(value = "ul")

public @ResponseBody

String multiFileUpload(HttpServletRequest request) throws IOException {

List<MultipartFile> files = ((MultipartHttpServletRequest) request).getFiles("fileName");

if (files.isEmpty()) {

return "上传失败！";

}

for (MultipartFile file : files) {

String fileName = file.getOriginalFilename();

int size = (int) file.getSize();

System.out.println(fileName + "-->" + size);

if (file.isEmpty()) {

return "上传失败！";

} else {

File dest = new File(uploadPath + "/upload/" + fileName);

if (!dest.getParentFile().exists()) { //判断文件父目录是否存在

dest.getParentFile().mkdir();

}

file.transferTo(dest);

}

}

return "上传成功！";

}

}

**启动**

<http://10.6.11.188:8088/ll>

