使用FTP对文件的上传和下载

# 一、上传、下载（可以是上传图片等，不建议使用）

1.使用ftp上传下载工具类

|  |
| --- |
| package com.sunshine.util;  import java.io.File;  import java.io.FileInputStream;  import java.io.FileNotFoundException;  import java.io.FileOutputStream;  import java.io.IOException;  import java.io.InputStream;  import java.io.OutputStream;  import org.apache.commons.net.ftp.FTP;  import org.apache.commons.net.ftp.FTPClient;  import org.apache.commons.net.ftp.FTPFile;  import org.apache.commons.net.ftp.FTPReply;  /\*\*  \* ftp上传下载工具类  \* <p>Title: FtpUtil</p>  \* <p>Description: </p>  \* <p>Company: www.itcast.com</p>  \* @author 入云龙  \* @date 2015年7月29日下午8:11:51  \* @version 1.0  \*/  public class FtpUtil {  /\*\*  \* Description: 向FTP服务器上传文件  \* @param host FTP服务器hostname  \* @param port FTP服务器端口  \* @param username FTP登录账号  \* @param password FTP登录密码  \* @param basePath FTP服务器基础目录  \* @param filePath FTP服务器文件存放路径。例如分日期存放：/2015/01/01。文件的路径为basePath+filePath  \* @param filename 上传到FTP服务器上的文件名  \* @param input 输入流  \* @return 成功返回true，否则返回false  \*/  public static boolean uploadFile(String host, int port, String username, String password, String basePath,  String filePath, String filename, InputStream input) {  System.out.println(host+"=host--port="+port);  boolean result = false;  FTPClient ftp = new FTPClient();  try {  int reply;  ftp.connect(host, port);// 连接FTP服务器  // 如果采用默认端口，可以使用ftp.connect(host)的方式直接连接FTP服务器  ftp.login(username, password);// 登录  reply = ftp.getReplyCode();  if (!FTPReply.isPositiveCompletion(reply)) {  ftp.disconnect();  return result;  }  //切换到上传目录  if (!ftp.changeWorkingDirectory(basePath+filePath)) {  //如果目录不存在创建目录  String[] dirs = filePath.split("/");  String tempPath = basePath;  for (String dir : dirs) {  if (null == dir || "".equals(dir)) continue;  tempPath += "/" + dir;  if (!ftp.changeWorkingDirectory(tempPath)) {  if (!ftp.makeDirectory(tempPath)) {  return result;  } else {  ftp.changeWorkingDirectory(tempPath);  }  }  }  }  //设置上传文件的类型为二进制类型  ftp.setFileType(FTP.BINARY\_FILE\_TYPE);  //上传文件  if (!ftp.storeFile(filename, input)) {  return result;  }  input.close();  ftp.logout();  result = true;  } catch (IOException e) {  e.printStackTrace();  } finally {  if (ftp.isConnected()) {  try {  ftp.disconnect();  } catch (IOException ioe) {  }  }  }  return result;  }    /\*\*  \* Description: 从FTP服务器下载文件  \* @param host FTP服务器hostname  \* @param port FTP服务器端口  \* @param username FTP登录账号  \* @param password FTP登录密码  \* @param remotePath FTP服务器上的相对路径  \* @param fileName 要下载的文件名  \* @param localPath 下载后保存到本地的路径  \* @return  \*/  public static boolean downloadFile(String host, int port, String username, String password, String remotePath,  String fileName, String localPath) {  boolean result = false;  FTPClient ftp = new FTPClient();  try {  int reply;  ftp.connect(host, port);  // 如果采用默认端口，可以使用ftp.connect(host)的方式直接连接FTP服务器  ftp.login(username, password);// 登录  reply = ftp.getReplyCode();  if (!FTPReply.isPositiveCompletion(reply)) {  ftp.disconnect();  return result;  }  ftp.changeWorkingDirectory(remotePath);// 转移到FTP服务器目录  FTPFile[] fs = ftp.listFiles();  for (FTPFile ff : fs) {  if (ff.getName().equals(fileName)) {  File localFile = new File(localPath + "/" + ff.getName());  OutputStream is = new FileOutputStream(localFile);  ftp.retrieveFile(ff.getName(), is);  is.close();  }  }  ftp.logout();  result = true;  } catch (IOException e) {  e.printStackTrace();  } finally {  if (ftp.isConnected()) {  try {  ftp.disconnect();  } catch (IOException ioe) {  }  }  }  return result;  }    public static void main(String[] args) {  try {  FileInputStream in=new FileInputStream(new File("H:\\AdminHome.psd"));  boolean flag = uploadFile("192.168.228.134", 21, "ftpuser", "ftpuser", "/home/ftpuser/www/images","/2015/01/21", "AdminHome.psd", in);  System.out.println(flag);  } catch (FileNotFoundException e) {  e.printStackTrace();  }  }  } |

1. 建立properties属性文件：

|  |
| --- |
| FTP\_ADDRESS=192.168.228.134  FTP\_PORT=21  FTP\_USERNAME=ftpuser  FTP\_PASSWORD=ftpuser  FTP\_BASE\_PATH=/home/ftpuser/www/images |

注释：FTP\_ADDRESS为FTP服务器的地址，FTP\_PORT为FTP服务器的端口号，FTP\_USERNAME为FTP服务器的用户名，FTP\_PASSWORD为FTP服务器的用户名的登录密码，FTP\_BASE\_PATH为上传文件到FTP服务器的路径

1. 在services中用spring的注解@value进行取出properties中对应的值，例如：

|  |
| --- |
| @Value("${FTP\_ADDRESS}")  **private** String ftp\_address;  @Value("${FTP\_PORT}")  **private** String ftp\_port;  @Value("${FTP\_USERNAME}")  **private** String ftp\_username;  @Value("${FTP\_PASSWORD}")  **private** String ftp\_password;  @Value("${FTP\_BASE\_PATH}")  **private** String ftp\_base\_path; |

1. 写相应的service方法：

|  |
| --- |
| **public** **boolean** upload(MultipartFile uploadFile) {  System.***out***.println(ftp\_address+"--"+ftp\_port+"--"+ftp\_username+"--"+ftp\_password+"--"+ftp\_base\_path);  String newName = "";  **boolean** result = **false**;  **try** {  //生成一个新的文件名  //取原始文件名  String oldName = uploadFile.getOriginalFilename();  System.***out***.println("oldname="+oldName);  //生成新文件名  //UUID.randomUUID();  newName = IDUtils.*genImageName*();  newName = newName+oldName.substring(oldName.lastIndexOf("."));  System.***out***.println("newname="+newName);    Date date = **new** Date();  SimpleDateFormat sim = **new** SimpleDateFormat("/yyyy/MM/dd");  String dateTime = sim.format(date);  //文件上传  result = FtpUtil.*uploadFile*(ftp\_address, **new** Integer(ftp\_port), ftp\_username, ftp\_password, ftp\_base\_path,  dateTime, newName, uploadFile.getInputStream());  } **catch** (Exception e) {    }  **return** result;  } |

1. 写相应的controller方法：

|  |
| --- |
| @RequestMapping("/uploadName")  **public** String upload(@RequestParam("fileName") CommonsMultipartFile upfile){  **boolean** result = **false**;  result = uploadService.upload(upfile);  **if**(result){  **return** "success";  }**else**{  **return** "error";  }  } |

Ok完成

至于ftp服务器的配置，看ftp服务器的配置文档。

# 上传、下载（可以传任何文件，建议使用）

1. 上传

|  |
| --- |
| **package** com.sunshine.service.impl;  **import** java.io.InputStream;  **import** java.io.OutputStream;  **import** java.util.HashMap;  **import** java.util.Map;  **import** java.util.UUID;  **import** javax.servlet.http.HttpServletResponse;  **import** org.apache.commons.net.ftp.FTP;  **import** org.apache.commons.net.ftp.FTPClient;  **import** org.joda.time.DateTime;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.beans.factory.annotation.Value;  **import** org.springframework.stereotype.Service;  **import** org.springframework.web.multipart.commons.CommonsMultipartFile;  **import** com.sunshine.service.UploadAndDowloadService;  @Service  **public** **class** UploadAndDowloadServiceImpl **implements** UploadAndDowloadService{    @Value("${FTP\_ADDRESS}")  **private** String ftp\_ip;  @Value("${FTP\_PORT}")  **private** **int** ftp\_port;  @Value("${FTP\_USERNAME}")  **private** String ftp\_name;  @Value("${FTP\_PASSWORD}")  **private** String ftp\_pass;  @Value("${FTP\_BASE\_PATH}")  **private** String ftp\_basepath;      **public** FTPClient connect() **throws** Exception {  // 创建一个FtpClient对象  FTPClient ftpClient = **new** FTPClient();  // 创建ftp连接,默认21端口  ftpClient.connect(ftp\_ip, ftp\_port);  // 登陆ftp服务器,使用用户名和密码  **boolean** bool = ftpClient.login(ftp\_name, ftp\_pass);  **if** (!bool)  **throw** **new** Exception("ftp连接失败");  **return** ftpClient;  }  /\*\*  \* ftp上传  \***@param** folder 文件上传到具体服务器上的目录下 如:invoice/excel  \***@param** file 上传的文件  \*/  @Override  **public** Map<String, String> upload(String folder, CommonsMultipartFile file) **throws** Exception {  Map<String, String> map = **new** HashMap<>();  String datePath = **new** DateTime().toString("/yyyy/MM/dd");  // 创建一个FtpClient对象  FTPClient ftpClient = connect();  InputStream fileInputStream = file.getInputStream();  // 设置上传路径  **if**(!folder.startsWith("/")) folder = "/"+folder;  **if**(folder.endsWith("/")) folder = folder.substring(0, folder.length()-1);  **if**(!ftpClient.changeWorkingDirectory(ftp\_basepath+folder+datePath)){  String[] forlderArr = (folder+datePath).split("/");  String tempPath = ftp\_basepath;  **for**(String fold :forlderArr){  **if**(**null**==fold ||"".equals(fold)) **continue**;  tempPath += "/" + fold;  **if** (!ftpClient.changeWorkingDirectory(tempPath)) {  **if** (!ftpClient.makeDirectory(tempPath)) {  **throw** **new** Exception("创建文件夹 : "+tempPath +"失败");  } **else** {  ftpClient.changeWorkingDirectory(tempPath);  }  }  }  }  // 修改上传文件格式 (二进制格式)  ftpClient.setFileType(FTP.***BINARY\_FILE\_TYPE***);  String UUIDName = **new** String(uuidName(file.getOriginalFilename()).getBytes(), "iso-8859-1");  // 参数1:服务器端文档名 , 参数2:上传文件的inputStream  **boolean** bool = ftpClient.storeFile( UUIDName, fileInputStream);  // 关闭连接  fileInputStream.close();  ftpClient.logout();  **if**(!bool){  delete(folder,UUIDName);  **throw** **new** Exception("ftp上传失败!");  }  map.put("folder", folder);  map.put("fileName", UUIDName);  **return** map;  }  // 生成uuid文件名  **private** String uuidName(String fileName) {  UUID uuid = UUID.*randomUUID*();  **return** uuid.toString() + "\_" + fileName;  }  @Autowired  **private** HttpServletResponse response;    /\*\*  \* ftp下载  \***@param** fileName 文件所在服务器路径 如:invoice/template.xlsx  \***@param** newName 下载到本地文件(中文)别名(不要后缀) 如:发票导入模板  \*/  @Override  **public** **void** download(String fileName,String newName) **throws** Exception {  // 创建一个FtpClient对象  FTPClient ftpClient = connect();  InputStream is = **null**;  OutputStream os = **null**;  **try** {  ftpClient.changeWorkingDirectory(ftp\_basepath);  is = ftpClient.retrieveFileStream(fileName);  **if** (is == **null**) {  is = ftpClient.retrieveFileStream(**new** String(fileName.getBytes(), "iso-8859-1"));  }  **if**(is==**null**)**throw** **new** Exception("下载链接失败!");  String end = fileName.substring(fileName.lastIndexOf("."));  response.setContentType("application/vnd.ms-excel");  response.setHeader("Content-Disposition",  "attachment;filename=" + **new** String((newName+end).getBytes("gbk"), "iso-8859-1"));  response.setHeader("Pragma", "public");  response.setHeader("Cache-Control", "max-age=0");    os = response.getOutputStream();  **byte**[] buffer = **new** **byte**[1024\*256];  **int** b = -1;  **while** ((b = is.read(buffer)) != -1) {  os.write(buffer, 0, b);  }  } **catch** (Exception e) {  **throw** e;  }**finally**{  **if**(is!=**null**){  is.close();  }  **if**(os!=**null**){  os.flush();  os.close();  }  ftpClient.logout();  }  }  /\*\*  \* ftp删除  \***@param** folder 文件所在服务器上的目录  \***@param** fileName 服务器上的文件名  \*/  @Override  **public** **boolean** delete(String folder, String fileName) **throws** Exception {  // 创建一个FtpClient对象  FTPClient ftpClient = connect();  **boolean** bool = **false**;  **if** (ftpClient != **null**) {  ftpClient.changeWorkingDirectory(ftp\_basepath + "/" + folder);  bool = ftpClient.deleteFile(fileName);  ftpClient.logout();  }  **return** bool;  }  } |