import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.util.Iterator;

import java.util.UUID;

import org.springframework.web.multipart.MultipartFile;

import org.springframework.web.multipart.MultipartHttpServletRequest;

public class UploadUtil {

/\*\*

\* 上传文件

\* @param muiltRequest

\* @param path

\* @return

\* @throws IOException

\*/

public static String uploadFile(MultipartHttpServletRequest muiltRequest,

String path) throws IOException {

// 获取遍历文件名

Iterator iter=muiltRequest.getFileNames();

String url=""; //返回的图片路径

while (iter.hasNext()) {

MultipartFile file=muiltRequest.getFile(iter.next().toString());

System.out.println("-->>>"+file);

if(!file.isEmpty()||file!=null) { //获取原始文件名

String filename = file.getOriginalFilename();

InputStream is = file.getInputStream(); // 获取输入流,MultipartFile中可以直接得到文件的流

int pos = filename.lastIndexOf("."); // 取文件的格式

//唯一标识数字

UUID uuid = UUID.randomUUID();

String filenameurl = path + '\\' + uuid + filename.substring(pos);

try {

// 获取输出流

OutputStream os = new FileOutputStream(filenameurl);

// 创建一个缓冲区

byte[] buffer = new byte[1024];

// 判断输入流中的数据是否已经读完的标识

int len = 0;

// 循环将输入流读入到缓冲区当中，(len=is.read(buffer))>0就表示is里面还有数据

while ((len = is.read(buffer)) > 0) {

// 使用FileOutputStream输出流将缓冲区的数据写入到指定的目录当中

os.write(buffer, 0, len);

}

os.flush();

os.close();

is.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

}

// url = url + uuid + filename.substring(pos)+";";

url = url + filenameurl+";";

}

}

url=url.substring(0, url.length()-1);

return url;

}