Atitit file process deduli file清理重复文件

/sumdoclist/src/comattilax/sumdoclist/clrDeliNameFile.java

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

// t1("D:\\l3 sumdoc s2018 torb31 v2 t1\_filelist.txt");

// trave\_dir("C:\\Users\\Administrator\\Documents\\law res 法学资源库","d:\\law res 法学资源库clrdeduli");

*trave\_dir*("D:\\BaiduNetdiskDownload","d:\\BaiduNetdiskDownload\_clrdeduli4");

// trave\_dir("C:\\Users\\Administrator\\Documents\\law res 法学资源库","d:\\law res 法学资源库clrdeduli3");

System.***out***.println("---");

}

**private** **static** **void** trave\_dir(String dir\_source, String dirout) **throws** Exception {

// 处理下级多层目录

Files.*walkFileTree*(Paths.*get*(dir\_source), **new** SimpleFileVisitor<Path>() {

@Override // 处理目录 dir

**public** FileVisitResult preVisitDirectory(Path dir, BasicFileAttributes attrs) **throws** IOException {

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

**return** FileVisitResult.***CONTINUE***;

}

// 处理文件

**public** FileVisitResult visitFile(Path file, BasicFileAttributes attrs) **throws** IOException {

// walkFile log

String fpath = file.toString();

**if**(fpath.contains("是印度最古老的一部法律文献"))

System.***out***.println("dbg");

String string = "\t正在访问" + fpath + "文件";

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

*logger*.info(string);

String ext=FilenameUtils.*getExtension*(file.toFile().getAbsolutePath());

String basenameBase=FilenameUtilsT55.*getBaseName*(file.toFile().getAbsolutePath());

String dirParent=file.toFile().getParent();

String baseFile = dirParent+"\\"+basenameBase+"."+ext;

*logger*.info("baseFile will"+baseFile);

String baseFile1 = dirParent+"\\"+basenameBase+"(1)."+ext;

Consumer<Map> fileConsumer1 = (paraM)->{

String rltPath=FilenameUtilsT55.*rltPath*(dirParent,dir\_source);

// String destFpath=dirout+"\\"+rltPath+"\\"+basenameBase+"("+i+")."+ext;

Map traceM=Maps.*newLinkedHashMap*();

traceM.put("act", "moveFile");

traceM.put("baseFile", paraM);

*logger*.info(JSON.*toJSONString*(traceM,**true**));

**try** {

//copyFile

FileUtils.*moveFile*((File)paraM.get("f"),(File)paraM.get("destFpath"));

} **catch** (Exception e) {

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

e.printStackTrace();

System.*exit*(0);

}

// duliFiles log

};

**if**(**new** File(baseFile).exists())

{

*process\_duliFiles*(**new** File(baseFile), **new** File(dirout), 1,50,dir\_source,fileConsumer1);

}**else**

{

**try** {

*process\_duliFiles\_byTop1*(**new** File(baseFile), **new** File(dirout), 50,dir\_source,fileConsumer1);

} **catch** (CantFindEx e) {

*logger*.error(e);

}

}

**return** FileVisitResult.***CONTINUE***; // 没找到继续找

}

});

//total log

// Map traceM=Maps.newLinkedHashMap();

// traceM.put("file", file);

// traceM.put("listSum", listSum);

// logger.info(JSON.toJSONString(traceM));

// 处理目录

}

**protected** **static** **void** process\_duliFiles\_byTop1(File baseFile, File destDir, **int** end, String baseDir,

Consumer<Map> fileConsumer1) **throws** CantFindEx, IOException {

String dir=baseFile.getParent();

String basenameBase=FilenameUtilsT55.*getBaseName*(baseFile.getAbsolutePath());

String ext=FilenameUtils.*getExtension*(baseFile.getAbsolutePath());

String rltPath=FilenameUtilsT55.*rltPath*(dir,baseDir);

**int** start;

start = *getStartIdex*(baseFile,end);

*process\_duliFiles*(baseFile, destDir, start+1, end, baseDir, fileConsumer1);

}

**protected** **static** **void** process\_duliFiles(File baseFile, File destDir, **int** start,**int** end, String baseDir,Consumer<Map> fileConsumer1) **throws** IOException {

**if**(baseFile.toString().contains("是印度最古老的一部法律文献"))

System.***out***.println("dbg");

String dir=baseFile.getParent();

String basenameBase=FilenameUtils.*getBaseName*(baseFile.getAbsolutePath());

String ext=FilenameUtils.*getExtension*(baseFile.getAbsolutePath());

String rltPath=FilenameUtilsT55.*rltPath*(dir,baseDir);

**for**(**int** i=start;i<end;i++)

{

String f=dir+"\\"+basenameBase+"("+i+")."+ext;

String destFpath=destDir+"\\"+rltPath+"\\"+basenameBase+"("+i+")."+ext;

**if**(**new** File(f).exists())

{

Map paramM=Maps.*newLinkedHashMap*();

paramM.put("f",**new** File( f)); paramM.put("destFpath",**new** File( destFpath));

paramM.put("start",start); paramM.put("end",end);paramM.put("curIdx",i);

fileConsumer1.accept(paramM);

}

/// FileUtils.moveFile(new File(f), new File(destFpath));

// FileUtils.moveFileToDirectory(new File(f), destDir, true);

}

// File[] subfiles=new File(dir).listFiles();

// for (File subf : subfiles) {

//

// }

}