

README

1. 只支持 Windows 文件系统。Linux，Mac 以及其他系统未经过测试。
2. 不支持中文路径。
3. 代码插入位置在 Explorer.java 的 run 函数中。可见注释如下。

```
//-----> Insert the file change CODE here<-----//
```

```
//-----END-----//
```

4. 所有的文件修改函数的说明

- 获取文件信息，传入参数均为 File 类型。均用 JAVA File 自带的方法。路径不正确的返回值与 File 的自带方法保持一致。

public synchronized String getName(File file)

public synchronized long getSize(File file)

public synchronized long getlastModified(File file)

- **public synchronized void** rename(File oldfile, String name)

重命名文件。

传入参数：

oldfile: 指向需要改名字的文件的File类。

name: 需要改的名字，必须为绝对路径。如: "C:\\Users*\\Desktop\\oo6\\aa.txt"

传入目录，文件不存在，路径错误会导致重命名失败。

- **public synchronized void** move(File file, String path)

移动文件。

传入参数：

file: 指向需要移动的文件File类。

path: 需要移动的路径，必须为绝对路径。如: "C:\\Users*\\Desktop\\oo6\\"

传入目录，文件不存在，路径错误会导致移动失败。

- **public synchronized void** addFile(String path)

创建一个文件

传入参数：

path: 加入文件的路径，必须为绝对路径。

如: "C:\\Users*\\Desktop\\oo6\\aa.txt"

传入目录，文件存在，会导致创建失败。

- **public synchronized void** addDir(String path)

创建一个文件夹

传入参数：

path: 加入文件夹的路径，必须为绝对路径。

如: "C:\\Users*\\Desktop\\oo6\\aa.txt"

传入文件，文件夹存在，会导致创建失败。

- **public synchronized void** deleteFile(String path)

删除文件，文件夹

传入路径

文件夹不可递归删除，只能删除空文件夹，否则会导致一次进行多次操作

- **public synchronized void** writefile(File file)

写入文件用于造成文件大小改变。

传入参数：

指向该文件的File类。

文件不存在，不是文件会导致写入失败。

5. 测试代码输入说明：

每输入一个文件/文件夹修改命令，请使用

```
try {
    Thread.sleep(15000);
} catch (InterruptedException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

休息15秒（给定频率）。

可以适当将扫描频率改小。但出错不负责任。

测试者更新文件的操作频率要小于被测试者给定的频率，同时每个工作区的总目录和文件数不要大于1000个。

6. 测试代码样例

```
for (int i = 0; i < 10; i++) {
    addFile("C:\\Users\\Aaa\\Desktop\\oo7\\" + i + "dd.txt");
    try {
        Thread.sleep(15000);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

for (int i = 0; i < 5; i++) {
    File file = new File("C:\\Users\\Aaa\\Desktop\\oo7\\" + i + "dd.txt");
    String string = "C:\\Users\\Aaa\\Desktop\\oo7\\cc\\";
    move(file, string);
    try {
        Thread.sleep(15000);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

writefile(new File("C:\\Users\\Aaa\\Desktop\\oo7\\aa.txt"));
try {
    Thread.sleep(15000);
} catch (InterruptedException e) {
    // TODO Auto-generated catch block
```

```

        e.printStackTrace();
    }

    File file = new File("C:\\Users\\Aaa\\Desktop\\oo7\\aa.txt");
    String string = "C:\\Users\\Aaa\\Desktop\\oo7\\cc\\";
    move(file, string);
    try {
        Thread.sleep(15000);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    rename(new File("C:\\Users\\Aaa\\Desktop\\oo7\\cc\\aa.txt"),
"C:\\Users\\Aaa\\Desktop\\oo7\\cc\\ii.txt");

    try {
        Thread.sleep(15000);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    deleteFile("C:\\Users\\Aaa\\Desktop\\oo7\\cc\\aa.txt");

```

7. 控制台输入

格式:

IF *path trigger* THEN *task*

...

run

path 为路径

trigger 可选 *renamed , modified , path_changed , size_changed*

task 可选 *record_summary , record_detail , recover*

run结束输入，输入过多会自动结束开始运行。

6个单词之间有且只能有一个空格

根据指导书，最多只能输入8条IFTTT命令。

*renamed , path_changed*路径只能是一个存在的文件。

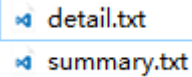
*recover*的触发器必须是 *renamed , path_changed*

样例:

IF C:\\Users\\Aaa\\Desktop\\oo7 modified THEN record_detail

```
IF C:\\Users\\Aaa\\Desktop\\oo7 modified THEN record_summary  
run
```

不可监视程序所在目录。



detail.txt
summary.txt

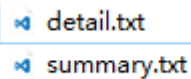
不可监视用于记录文件详细信息的一个文件。即

8. 逻辑细节说明

- A. 文件夹里所包含的任一文件size变化，新增文件，删除文件，均认为该文件夹size出现变化，即触发size_changed。
- B. 文件夹里所包含的任一文件size变化，新增文件，删除文件，以及文件被修改，均认为该文件夹被修改，即触发modified。
- C. 只有触发器对应的任务包含record_summary时，才会增加触发器的触发次数记录。否则不会启动。

9. 输出说明

位于工程文件根目录下的



detail.txt
summary.txt

detail.txt

记录所有trigger的触发情况以及对应的变化。

例：

Trigger: size_changed ADD-> Odd.txt and all of its father folder's size changed.

说明：文件夹的size改变， 因为新增了 Odd.txt 文件。

Trigger: size_changed DELETE-> aa.txt and all of its father folder's size changed.

说明：文件夹的size改变， 因为删除了 aa.txt 文件。

Trigger: size_changed aa.txt 0->3

说明：文件aa.txt 的size改变， 0->3。

Trigger: modified aa.txt 1460735003771->1460781266683 and all of its father folder's modified time.

说明：文件夹的修改时间改变， 因为其中的aa.txt文件被修改。

summary.txt

记录次数

PS： 请等待足够长的时间

感谢同学耐心的读完readme~~~~

无忌拍脑瓜想出来的作业实在太傻逼了。

周末愉快，学业有成~~~