

Path 操作

1.删除文件

```
public static void main(String[] args) {  
    Path path = Paths.get("H://afile");// 文件不存在  
    try {  
        Files.delete(path); // 删除一个不存在的文件  
    } catch (IOException e) {  
        System.out.println(e.getMessage());  
    }  
}
```

2.遍历目录,不包括子目录的文件

```
Path dir = Paths.get("F:\\LZ\\pdf");  
DirectoryStream<Path> stream = Files.newDirectoryStream(dir);  
for(Path path : stream){  
    System.out.println(path.getFileName());  
}
```

2.遍历目录,及其子目录下的文件

```
Path dir = Paths.get("F:\\LZ\\pdf");  
Files.walkFileTree(dir,new SimpleFileVisitor<Path>(){  
    @Override  
    public FileVisitResult visitFile(Path file, BasicFileAttributes attrs)  
        throws IOException {  
        if(file.toString().endsWith(".pdf")){  
            System.out.println(file.getFileName());  
        }  
        return super.visitFile(file, attrs);  
    }  
});
```

3.创建多级目录

```
Path dir = Paths.get("F:\\LZ\\xx\\dd");  
Files.createDirectories(dir);
```

4.创建文件, 不存在则抛出异常

```
Path dir = Paths.get("F:\\LZ\\xx\\dd\\1.txt");
Files.createFile(dir);
```

5.文件的复制

```
Path src = Paths.get("F:\\LZ\\xx\\dd\\1.txt");
Path target = Paths.get("F:\\LZ\\xx\\dd\\2.txt");

//REPLACE_EXISTING:文件存在，就替换
Files.copy(src, target,StandardCopyOption.REPLACE_EXISTING);
```

6.一行一行读取文件

```
Path src = Paths.get("F:\\LZ\\xx\\dd\\1.txt");
BufferedReader reader = Files.newBufferedReader(src,StandardCharsets.UTF_8);
String line ;
while((line=reader.readLine()) != null){
    System.out.println(line);
}
reader.close();
```

7.写入字符串

```
Path src = Paths.get("F:\\LZ\\xx\\dd\\1.txt");
BufferedWriter writer = Files.newBufferedWriter(src, StandardCharsets.UTF_8
    ,StandardOpenOption.APPEND);//追加
writer.write("hello word Path");
writer.close();
```

8.二进制 读写 与字符串类似

1. 一个方法直接去取字符串 和 二进制流

```
// 字符串
Path src = Paths.get("F:\\LZ\\xx\\dd\\1.txt");
for(String line : Files.readAllLines(src)){
    System.out.println(line);
}

// 二进制流
byte[] bytes = Files.readAllBytes(src);
```

10.监测是否 目录下的 文件 , 目录 被修改 , 创建 , 或删除.

```
Path dir = Paths.get("F:\\LZ\\xx\\dd"); // 必须是目录
WatchService service = FileSystems.getDefault().newWatchService();
WatchKey key = dir.register(service, StandardWatchEventKinds.ENTRY_MODIFY //文件修改
,StandardWatchEventKinds.ENTRY_DELETE //文件 被删除
,StandardWatchEventKinds.ENTRY_CREATE); //文件被创建
while(true){
    key = service.take();
    for(WatchEvent<?> event : key.pollEvents()){
        if(event.kind() == StandardWatchEventKinds.ENTRY_MODIFY){ // 文件被修改了
            System.out.println("这个目录下的文件被修改了");
        }else if(event.kind() == StandardWatchEventKinds.ENTRY_DELETE){
            System.out.println("这个目录下的文件被删除了");
        }else if(event.kind() == StandardWatchEventKinds.ENTRY_CREATE){
            System.out.println("这个目录下的文件被创建了");
        }
    }
    key.reset();
}
```

11.读取文件的 最后 1000 个字符

```
Path dir = Paths.get("F:\\LZ\\xx\\dd\\1.txt"); // 必须是目录
ByteBuffer buffer = ByteBuffer.allocate(1024);
FileChannel channel = FileChannel.open(dir, StandardOpenOption.READ);
channel.read(buffer,channel.size() - 1000);
System.out.println(buffer);
```