软件复用课程设计

Client-Server 应用程序

程序文档3.0

小组： Team10

成员： 谢志杰 1352975

计鹏玥 1352914

王思尧 1352896

谢明玥 1352937

徐锦程 1353012

文档变更记录

|  |  |  |
| --- | --- | --- |
| 更改人 | 日期 | 更改内容 |
| 计鹏玥 | 2016年5月2日 | 创建文档 |
| 计鹏玥，谢明玥 | 2016年5月3日 | 接口说明 |
| 计鹏玥 | 2016年5月5日 | 构件修改 |
| 计鹏玥 | 2016年5月10日 | 添加对输出文件的控制及加密 |

[文档变更记录 1](#_Toc2588)

[1．开发规划 1](#_Toc17838)

[1.1开发人员 1](#_Toc20021)

[1.2开发计划 1](#_Toc15385)

[1.3开发环境和工具 1](#_Toc28783)

[2．功能扩展 2](#_Toc23580)

[2.1 File 2](#_Toc14077)

[2.1.1 功能扩展说明 2](#_Toc25808)

[2.1.2 扩展方案 2](#_Toc5987)

[2.1.3 问题&解决 2](#_Toc9528)

[2.2 PackerUtils 3](#_Toc15866)

[2.2.1 功能扩展说明 3](#_Toc21539)

[2.2.2 扩展方案 3](#_Toc15048)

[2.2.3 问题&解决 3](#_Toc27688)

[3．接口规范 3](#_Toc17934)

[3.1 File 3](#_Toc21632)

[3.1.1 SaveToFile.java 3](#_Toc25132)

[3.1.2 FileUtils.java 4](#_Toc9682)

[3.2 PackerUtils 5](#_Toc3539)

[3.2.1 Packer.java 5](#_Toc4645)

[3.2.2 PackPerDay.java 6](#_Toc4254)

[3.2.3 PackPerWeek.java 6](#_Toc29076)

[3.2.4 Unpacker.java 6](#_Toc32129)

[3.2.5 DESEncryptor.java 6](#_Toc16763)

1．开发规划

1.1开发人员

|  |  |
| --- | --- |
| 学号 | 姓名 |
| 1352975 | 谢志杰 |
| 1352914 | 计鹏玥 |
| 1352896 | 王思尧 |
| 1352937 | 谢明玥 |
| 1353012 | 徐锦程 |

1.2开发计划

|  |  |  |
| --- | --- | --- |
| 起始时间 | 结束时间 | 计划 |
| 2016年4月20日 | 2016年4月27日 | 基于选择的组件进行开发，保存所有收到的消息到文件，并将每天所有的输出文件归档成一个压缩包。 |
| 2016年4月27日 | 2016年5月4日 | 对文件进行输出和压缩，配置同组成员，并只向组内成员广播消息；将每周的七个归档文件重新生成一个压缩包。 |
|  |  |  |
|  |  |  |

1.3开发环境和工具

GitHub - 版本管理工具

Java - 开发语言

JUnit - 单元测试框架

RabbitMQ - 中间件

2．功能扩展

2.1 File

2.1.1 功能扩展说明

对输出文件进行控制：

1. 显示输出文件的时间
2. 对输出文件的大小进行限制
3. 对总的文件大小进行限制
4. 可以对输出文件的参数进行配置

2.1.2 扩展方案

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **时间** | **客户端** | **服务端** |
| 1 | 5.5 - 5.10 | 输出文件控制使用PM构件提供方team1提供的PM构件新增功能来实现。 | 该功能使用team1提供的PM完成。team1很及时的更新了PM的接口，提供了相应的接口来设置单个文件最大值（setMaxFileSize）和所有文件最大值（setMaxTotalSize）。 |

2.1.3 问题&解决

|  |  |  |
| --- | --- | --- |
| **No.** | **问题** | **解决** |
| 1 | 第一次使用我们发现了一个漏洞，即SizeUnit被protected修饰，导致无法在package以外使用。 | 向team1反馈后很快修复。 |
| 2 | 在使用过程中，PM的表现很异常。具体为，文件大小上限并不准确，我们设置了500MB的总文件上限，但是仅仅产生了10个文件以后，文件夹就被清空了，然后不断重复这个过程。 | 在确定这是一个构件本身的漏洞以后，我们会向team1汇报。 |
| 3 | 在client端的使用team1的构件的过程中，我们发现了一些问题。 | 其中有关SizeUnit的问题得到了解决。 |
| 4 | 在构件替换这一功能扩展中，我们发现client端的一些与PM、CM有关的功能仍然在使用我们小组自己实现的构件。 | 这次完全替换为PM、CM构件提供方提供的构件。 |

2.2 PackerUtils

2.2.1 功能扩展说明

1. 对归档文件进行加密。
2. 可以将加密后的进行解密

2.2.2 扩展方案

我们通过扩充PackerUtils构件实现了这一扩展功能。

一个新的类：DESEncryptor被用于提供DES加密服务。

Packer能够在压缩的时候选择是否进行加密，加密会在压缩后进行，具体原因如下：

（1）先压缩后加密：运算量小一些。

（2）加密后压缩率可能会受到影响。

注：归档文件加密功能目前集成于文件打包Packer构件中，因此客户端的有关接口无需更改便可实现加密功能。

2.2.3 问题&解决

|  |  |  |
| --- | --- | --- |
| **No.** | **问题** | **解决** |
| 1 | 如果进行了合适的加密，加密后的数据通常为随机数据，而大多数的压缩算法通过寻找数据中的pattern来进行压缩，但是加密后可能就破坏了这种数据的规律性，导致数据无法进行压缩。 |  |
| 2 | 目前在考虑这种实现的局限性，即打包功能未向外提供有关加密选择的接口。 | 需要进一步完善Packer构件的功能。 |

3．接口规范

3.1 File

3.1.1 SaveToFile.java

(1) SaveToFile()

获取配置信息所指定的路径。

(2) SaveToFile(String s)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| s | String | 文件路径 |

(3) getBufferedWriter()

获取输出流，并创建文件。

(4) write(String str)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| str | String | 准备写入文件的内容 |

(5) getDirectoryPath()

|  |  |  |
| --- | --- | --- |
| 返回值 | 类型 | 含义 |
| directoryPath | String | 文件夹路径 |

(6) finalized()

关闭文件输入流。

3.1.2 FileUtils.java

(1) filenameWithoutExt(String fileNameWithExt)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| fileNameWithExt | String | 包含扩展文件名的文件名称 |

|  |  |  |
| --- | --- | --- |
| 返回值 | 类型 | 含义 |
| fileNameWithoutExt | String | 不包含扩展文件名的文件名称 |

(2) renameFileExtension(String source, String newExtension)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| source | String | 文件源 |
| newExtension | String | 新的扩展文件名 |

|  |  |  |
| --- | --- | --- |
| 返回值 | 类型 | 含义 |
| File(source).renameTo(new File(target)) | Boolean | 判断是否对文件扩展名重命名成功 |

(3) getFileExtension(String f)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| f | String | 最近的文件 |

|  |  |  |
| --- | --- | --- |
| 返回值 | 类型 | 含义 |
| ext | String | 新的扩展文件名 |

3.2 PackerUtils

3.2.1 Packer.java

(1) Packer(String ipath, String opath)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| ipath | String | 文件读取路径 |
| opath | String | 文件保存路径 |

1. packupSuffix(String suffix, boolean encryptIt)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| suffix | String | 文件后缀名称 |
| encryptIt | Boolean | 判断文件是否要进行加密 |

(3) addToZipFile(File file, ZipOutputStream zos)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| file | File | 文件 |
| zos | ZipOutputStream | 文件输出为压缩的zip格式 |

1. PackerTimer(String logFilename)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| logFilename | String | 日志文件的名字 |

3.2.2 PackPerDay.java

1. PackPerDay(String iPath, String oPath)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| iPath | string | 文件读取路径 |
| oPath | string | 文件保存路径 |

3.2.3 PackPerWeek.java

(1) PackPerWeek(String iPath, String oPath)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| iPath | string | 文件读取路径 |
| oPath | string | 文件保存路径 |

3.2.4 Unpacker.java

(1) unZip(String zipFile, String outputFolder)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| zipFile | String | 压缩文件的文件名 |
| outputFolder | String | 文件的解压路径 |

3.2.5 DESEncryptor.java

(1) DESEncryptor()

调用函数generateKey()来生成密钥。

(2) DESEncryptor(SecretKey ikey)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| ikey | SecretKey | 文件的密钥 |

(3) generateKey()

|  |  |  |
| --- | --- | --- |
| 返回值 | 类型 | 含义 |
| generateKey() | SecretKey | 生成的文件密钥 |

(4) saveKey(SecretKey key, File file)

|  |  |  |
| --- | --- | --- |
| 返回值 | 类型 | 含义 |
| key | SecretKey | 文件密钥 |
| file | File | 密钥对应的文件 |

(5) loadKey(File file)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| file | File | 密钥对应的文件 |

|  |  |  |
| --- | --- | --- |
| 返回值 | 类型 | 含义 |
| SecretKeySpec(encoded, "DES") | SecretKey | 采用DES算法加密后的密钥 |

(5) encrypt(String file, String destFile)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| file | String | 将要加密的文件 |
| destFile | String | 加密后的文件 |

(6) decrypt(String file, String dest)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 含义 |
| file | File | 将要解密的文件 |
| dest | String | 解密后文件 |