

## Java程序员的日常—— IOUtils总结

以前写文件的复制很麻烦，需要各种输入流，然后读取line，输出到输出流...其实apache.commons.io里面提供了输入流输出流的常用工具方法，非常方便。下面就结合源码，看看IOUtils都有什么用处吧！

代码参考<https://github.com/xinghalo/JDK-Learning>

### 常用的静态变量

在IOUtils中还是有很多常用的一些变量的，比如换行符等等

```
public static final char DIR_SEPARATOR_UNIX = '/';
public static final char DIR_SEPARATOR_WINDOWS = '\\';
public static final char DIR_SEPARATOR;
public static final String LINE_SEPARATOR_UNIX = "\n";
public static final String LINE_SEPARATOR_WINDOWS = "\r\n";
public static final String LINE_SEPARATOR;

static {
    DIR_SEPARATOR = File.separatorChar;

    StringBuilderWriter buf = new StringBuilderWriter(4);
    PrintWriter out = new PrintWriter(buf);
    out.println();
    LINE_SEPARATOR = buf.toString();
    out.close();
}
```

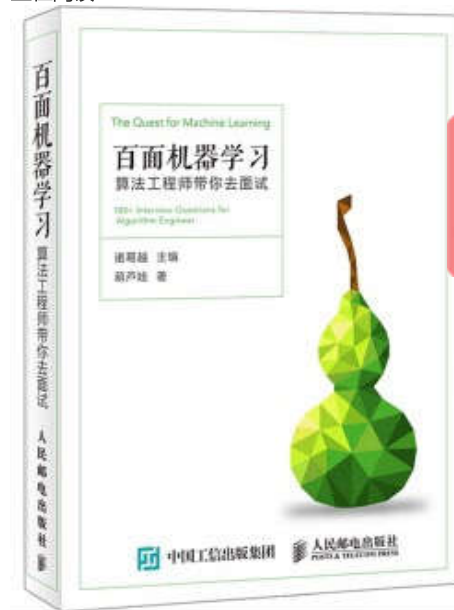
### 常用方法

#### copy

这个方法可以拷贝流，算是这个工具类中使用最多的方法了。支持多种数据间的拷贝：

#### 公告

正在阅读...



昵称: xingoo

园龄: 6年1个月

粉丝: 3358

关注: 79

[+加关注](#)

| 2018年10月          |    |                    |    |                    |    |                    |
|-------------------|----|--------------------|----|--------------------|----|--------------------|
| 日                 | 一  | 二                  | 三  | 四                  | 五  | 六                  |
| 30                | 1  | 2                  | 3  | 4                  | 5  | 6                  |
| <a href="#">7</a> | 8  | 9                  | 10 | <a href="#">11</a> | 12 | 13                 |
| 14                | 15 | <a href="#">16</a> | 17 | 18                 | 19 | <a href="#">20</a> |
| 21                | 22 | 23                 | 24 | 25                 | 26 | 27                 |
| 28                | 29 | 30                 | 31 | 1                  | 2  | 3                  |
| 4                 | 5  | 6                  | 7  | 8                  | 9  | 10                 |

#### 最新随笔

```
copy(inputStream,outputstream)
copy(inputStream,writer)
copy(inputStream,writer,encoding)
copy(reader,outputstream)
copy(reader,writer)
copy(reader,writer,encoding)
```

copy内部使用的其实还是copyLarge方法。因为copy能拷贝Integer.MAX\_VALUE的字节数据，即 $2^{31}-1$ 。

### copyLarge

这个方法适合拷贝较大的数据流，比如2G以上。

```
copyLarge(reader,writer) 默认会用1024*4的buffer来读取
copyLarge(reader,writer,buffer)
```

内部的细节可以参考：

```
public static long copyLarge(Reader input, Writer output, char [] buffer) throws IOException {
    long count = 0;
    int n = 0;
    while (EOF != (n = input.read(buffer))) {
        output.write(buffer, 0, n);
        count += n;
    }
    return count;
}
```

这个方法会用一个固定大小的Buffer，持续不断的读取数据，然后写入到输出流中。

### read

从一个流中读取内容

```
read(inputStream,byte[])
read(inputStream,byte[],offset,length)
//offset是buffer的偏移值，length是读取的长度

read(reader,char[])
read(reader,char[],offset,length)
```

这里我写了个小例子，可以测试read方法的效果：

1. 《影响力》—— 读后总结
2. 《Opencv 3 计算机视觉 python语言实现》· 第二遍 —— 读后笔记
3. 程序猿的日常——工作中常用的Shell脚本
4. 《放学后》—— 读后总结
5. 《美团机器学习实践》—— 读后总结
6. MSRA-TD5000数据集使用详解
7. 2017CS231n学习笔记——计算机视觉的概述
8. Java几种单例模式的实现与利弊
9. python项目实现配置统一管理的方法
10. 我的AI之路 —— OCR文字识别快速体验版

随笔分类(881)

AngularJS(27)  
Elasticsearch(53)  
Hadoop(18)  
Hbase(5)  
Java(156)  
JavaScript(71)  
Kafka(3)  
Keras(1)  
linux(30)  
Logstash(16)  
Mac说(2)  
MongoDB(2)  
Oozie(12)  
Oracle(44)  
Python(3)  
Redis(1)  
Ruby(8)  
Scala(8)  
Spark(35)  
Spring(25)  
Sqoop(3)  
TensorFlow(1)  
Zookeeper(4)  
程序人生(128)  
缓存系统(6)  
机器学习(32)  
全栈折腾(11)  
软件考试(1)  
设计模式(23)  
数据仓库(9)  
数学理论(3)

赞  
助

```
@Test
public void readTest(){
    try{
        byte[] bytes = new byte[4];
        InputStream is = IOUtils.toInputStream("hello world");
        IOUtils.read(is, bytes);
        System.out.println(new String(bytes));

        bytes = new byte[10];
        is = IOUtils.toInputStream("hello world");
        IOUtils.read(is, bytes, 2, 4);
        System.out.println(new String(bytes));
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

得到的结果是：

```
hell
[]hell[]
```

### readFully

这个方法会读取指定长度的流，如果读取的长度不够，就会抛出异常

```
readFully(inputstream,byte[])
readFully(inputstream,byte[],offset,length)
readFully(reader,charp[])
readFully(reader,char[],offset,length)
```

比如：

```
@Test
public void readFullyTest(){
    byte[] bytes = new byte[4];
    InputStream is = IOUtils.toInputStream("hello world");
    try {
        IOUtils.readFully(is,bytes);
        System.out.println(new String(bytes));
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

算法(86)  
图像处理(4)  
网络(20)  
杂谈(24)  
自然语言处理(6)

### 常用工具

leetcode  
processon  
stackedit

### 官方文档

keras  
mybatis  
nvidia  
opencv3.4.2  
opencv-python  
python3.6  
tomcat

### 好友链接

huaxiaozhuan  
李震

### 积分与排名

积分 - 1481154  
排名 - 41

赞  
助

输出

```
hell
```

但是如果读取20个byte,就会出错了

```
java.io.EOFException: Length to read: 20 actual: 11
    at org.apache.commons.io.IOUtils.readFully(IOUtils.java:2539)
    at org.apache.commons.io.IOUtils.readFully(IOUtils.java:2558)
    at test.java.IOUtilsTest.readFullyTest(IOUtilsTest.java:22)
    ...
```

### readLines

readLines方法可以从流中读取内容,并转换为String的list

```
readLines(inputStream)
readLines(inputStream, charset)
readLines(inputStream, encoding)
readLines(reader)
```

这个方法极大简化了之前原始的读取方法:

```
@Test
public void readLinesTest() {
    try {
        InputStream is = new FileInputStream("D://test1.txt");
        List<String> lines = IOUtils.readLines(is);
        for (String line : lines) {
            System.out.println(line);
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

输出内容:

```
hello
world
```

赞助

```
nihao
ioutils
```

## skip

这个方法用于跳过指定长度的流,

```
skip(inputStream, skip_length)
skip(ReadableByteChannel, skip_length)
skip(reader, skip_length)
```

例如:

```
@Test
public void skipTest() {
    InputStream is = IOUtils.toInputStream("hello world");
    try {
        IOUtils.skip(is, 4);
        System.out.println(IOUtils.toString(is, "utf-8"));
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

## skipFully

这个方法类似skip, 只是如果忽略的长度大于现有的长度, 就会抛出异常

```
skipFully(inputStream, toSkip)
skipFully(readableByteChannel, toSkip)
skipFully(inputStream, toSkip)
```

例如

```
@Test
public void skipFullyTest() {
    InputStream is = IOUtils.toInputStream("hello world");
    try {
        IOUtils.skipFully(is, 30);
        System.out.println(IOUtils.toString(is, "utf-8"));
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

赞助

## write

这个方法可以把数据写入到输出流中

```
write(byte[] data, OutputStream output)
write(byte[] data, Writer output)
write(byte[] data, Writer output, Charset encoding)
write(byte[] data, Writer output, String encoding)
write(char[] data, OutputStream output)
write(char[] data, OutputStream output, Charset encoding)
write(char[] data, OutputStream output, String encoding)
write(char[] data, Writer output)
write(CharSequence data, OutputStream output)
write(CharSequence data, OutputStream output, Charset encoding)
write(CharSequence data, OutputStream output, String encoding)
write(CharSequence data, Writer output)
write(StringBuffer data, OutputStream output)
write(StringBuffer data, OutputStream output, String encoding)
write(StringBuffer data, Writer output)
write(String data, OutputStream output)
write(String data, OutputStream output, Charset encoding)
write(String data, OutputStream output, String encoding)
write(String data, Writer output)
```

例如

```
@Test
public void writeTest() {
    try {
        OutputStream os = new FileOutputStream("E:/test.txt");
        IOUtils.write("hello write!", os);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

## writeLines

这个方法可以把string的List写入到输出流中

```
writeLines(Collection<?> lines, String lineEnding, OutputStream output)
writeLines(Collection<?> lines, String lineEnding, OutputStream output, Charset encoding)
```

赞助

```
writeLines(Collection<?> lines, String lineEnding, OutputStream output, String encoding)
writeLines(Collection<?> lines, String lineEnding, Writer writer)
```

例如

```
@Test
public void writeLinesTest() throws IOException {
    List<String> lines = new ArrayList();
    lines.add("hello");
    lines.add("list");
    lines.add("to");
    lines.add("file");
    OutputStream os = new FileOutputStream("E:/test.txt");
    IOUtils.writeLines(lines, IOUtils.LINE_SEPARATOR, os);
}
```

## close

关闭URL连接

```
close(URLConnection conn)
```

## closeQuietly

忽略nulls和异常，关闭某个流

```
close(URLConnection conn)
closeQuietly(Closeable... closeables)
closeQuietly(Closeable closeable)
closeQuietly(InputStream input)
closeQuietly(OutputStream output)
closeQuietly(Reader input)
closeQuietly(Selector selector)
closeQuietly(ServerSocket sock)
closeQuietly(Socket sock)
closeQuietly(Writer output)
```

## contentEquals

比较两个流是否相同

```
contentEquals(InputStream input1, InputStream input2)
contentEquals(Reader input1, Reader input2)
```

赞助

例如

```
@Test
public void contentEqualsTest(){
    InputStream is1 = IOUtils.toInputStream("hello123");
    InputStream is2 = IOUtils.toInputStream("hello123");

    try {
        System.out.println(IOUtils.contentEquals(is1,is2));
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

### contentEqualsIgnoreEOL

比较两个流，忽略换行符

```
contentEqualsIgnoreEOL(Reader input1, Reader input2)
```

### lineIterator

读取流，返回迭代器

```
lineIterator(InputStream input, Charset encoding)
lineIterator(InputStream input, String encoding)
lineIterator(Reader reader)
```

### toBufferedInputStream

把流的全部内容放在另一个流中

```
toBufferedInputStream(InputStream input)
toBufferedInputStream(InputStream input, int size)
```

### toBufferedReader

返回输入流

```
toBufferedReader(Reader reader)
toBufferedReader(Reader reader, int size)
```

### toByteArray

赞助



## 返回字节数组

```
toByteArray(InputStream input)
toByteArray(InputStream input, int size)
toByteArray(InputStream input, long size)
toByteArray(Reader input)
toByteArray(Reader input, Charset encoding)
toByteArray(Reader input, String encoding)
toByteArray(String input)
toByteArray(URL uri)
toByteArray(URL url)
toByteArray(URLConnection urlConn)
```

## toCharArray

## 返回字符数组

```
toCharArray(InputStream is)
toCharArray(InputStream is, Charset encoding)
toCharArray(InputStream is, String encoding)
toCharArray(Reader input)
```

## toInputStream

## 返回输入流

```
toInputStream(CharSequence input)
toInputStream(CharSequence input, Charset encoding)
toInputStream(CharSequence input, String encoding)
toInputStream(String input)
toInputStream(String input, Charset encoding)
toInputStream(String input, String encoding)
```

## toString

## 返回字符串

```
toString(byte[] input)
toString(byte[] input, String encoding)
toString(InputStream input)
toString(InputStream input, Charset encoding)
toString(InputStream input, String encoding)
toString(Reader input)
toString(URL uri)
toString(URL uri, Charset encoding)
```

赞助

```
toString(Uri uri, String encoding)
toString(URL url)
toString(URL url, Charset encoding)
toString(URL url, String encoding)
```

作者: [xingoo](#)

出处: <http://www.cnblogs.com/xing901022>

本文版权归作者和博客园共有。欢迎转载，但必须保留此段声明，且在文章页面明显位置给出原文连接！

分类: [Java](#)

标签: [IOUtils](#)

好文要顶

关注我

收藏该文



xingoo



关注 - 79

粉丝 - 3358

2

0

+加关注

« 上一篇: [sql基础知识: 分页+排序](#)

» 下一篇: [如何把IP转换成经纬度 \(Java版\)](#)

posted @ 2016-10-19 22:23 [xingoo](#) 阅读(20136) 评论(1) [编辑](#) [收藏](#)

评论列表

#1楼 2017-05-31 13:56 [天涯已可](#)

瞬间懵逼.....

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

赞助

