

Daily_Learning(2019-12-4)

使用node.js实现读写文件操作，使用promise封装读写函数。并且实现从一个input.txt文件读取一串逗号分开的数字，排序后，写入到output.txt文件。

```
1 let fs = require("fs");
2
3
4 /**
5  * 封装异步读取文件
6  * @param filePath 文件路径
7  * @return promise
8  */
9 const readFile = (filePath) => {
10     return new Promise((reslove, reject) => {
11         fs.readFile(filePath, 'utf8', (err, data) => {
12             if (!err) {
13                 reslove(data)
14             } else {
15                 reject(err)
16             }
17         })
18     })
19 }
20
21
22 /**
23  * 封装异步写入文件
24  * @param filePath 文件路径
25  * @param data 写入的数据
26  * @return promise
27  */
28 const writeFile = (filePath, data) => {
29     return new Promise((reslove, reject) => {
30         fs.writeFile(filePath, JSON.stringify(data), (err) => {
31             if (!err) {
32                 reslove(data)
33             } else {
34                 reject(err)
35             }
36         })
37     })
38 }
```

```
38 }
39 /**
40  * 冒泡排序
41  * @param arr
42  * @return arr
43  */
44
45
46 function bubbleSort(arr) {
47     var len = arr.length;
48     for (var i = 0; i < len; i++) {
49         for (var j = 0; j < len - 1 - i; j++) {
50             if (arr[j] > arr[j+1]) {
51                 var temp = arr[j+1];
52                 arr[j+1] = arr[j];
53                 arr[j] = temp;
54             }
55         }
56     }
57     return arr;
58 }
59
60
61 /**
62  * 读取input文件中逗号相隔的数字，并排序
63  * @param filePathForRead 文件读取路径
64  * @param filePathForWrite 文件写入路径
65  * @return promise
66  */
67 const handelData = async (filePathForRead,filePathForWrite)=>{
68     //读取数据并转换为数字数组。
69     const readData = await readFile(filePathForRead);
70     const readDataArr = readData.split(',').map( ele=> +ele );
71     // 冒泡排序
72     const data = bubbleSort(readDataArr).join('');
73     // 写入数据
74     const output = await writeFile(filePathForWrite,data);
75     console.log(output)
76 }
77
78
79
80 handelData('./input.txt','./output.txt');
```