

# Nodejs对Excel文件的操作

nodejs平台可以借用于第三方的模块去操作excel,它使用的模块就是 node-xlsx

## 创建项目

首先使用nodejs创建项目，初始化项目，然后安装所需要的第三方模块

```
$ npm init --yes
```

接下来安装依赖

```
$ npm install node-xlsx --save
```

## 准备excel文件

随便到一个excel文件进行读取即可。格式如下

#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
#	sid	sname	ssex	sbirthday	snation	slDCard	saddr	smail	stel	isDel	cid	sphoto	spwd	slocal	stime
1	2005010101	苏俊丹	女	1989/9/1	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005010101@softeem.com	13878841001	2			123456	ffff127.0.0.1	2019/11/7 21:43
2	2005010102	张小苗	男	1989/9/2	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005010102@softeem.com	13878941002	2			123456		
3	2005010201	赵希坤	男	1989/9/3	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005010201@softeem.com	13879041003	3			123456		
4	2005010202	阮志婷	女	1989/9/4	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005010202@softeem.com	13879141004	3			123456		
5	2005020101	程丽婷	男	1989/9/5	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005020101@softeem.com	13879241005	3			123456		
6	2005020102	成正韦	男	1989/9/5	回族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005020102@softeem.com	13879341006	3			147258	ffff127.0.0.1	2019/11/7 20:35
7	2005020201	李家洋	男	1989/9/7	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005020201@softeem.com	13879441007	1			123456		
8	2005020202	孟彬彬	女	1989/9/8	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005020202@softeem.com	13879541008	1			123456		
9	2005030101	蔡莎莎	女	1989/9/9	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005030101@softeem.com	13879641009	2			123456		
10	2005030102	蔡金奎	男	1989/9/10	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005030102@softeem.com	13879741010	1			123456		
11	2005040101	陈亚克	女	1989/9/11	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005040101@softeem.com	13879841011	4			123456		
12	2005040102	苏晋秀	女	1989/9/12	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005040102@softeem.com	13879941012	1			123456		
13	2005040201	蔡晋勇	男	1989/9/13	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005040201@softeem.com	13880041013	4			123456		
14	2005040202	侯志方	女	1989/9/14	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005040202@softeem.com	13880141014	1			123456		
15	2005050101	李明	男	1989/9/15	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005050101@softeem.com	13880241015	4			123456		
16	2005050102	陈曦	女	1989/9/16	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005050102@softeem.com	13880341016	1			123456		
17	2005050201	郑小茜	男	1989/9/17	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005050201@softeem.com	13880441017	3			123456		
18	2005050202	王静	女	1989/9/18	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005050202@softeem.com	13880541018	1			123456		
19	2005050208	余珊珊	女	1989/9/19	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005050208@softeem.com	13880641019	2			123456		
20	2005050209	杨标	男	1989/9/20	汉族	4.20984E+17	湖北省/武汉市/武昌区/光谷广场1号	2005050209@softeem.com	13880741020	1			123456		
21	5.163E+11	杨标	女	2019/10/10	阿昌族	4.20222E+17	湖北省/武汉市/武昌区/光谷广场1号	www@qq.com	1224785544	4	/uploadimg		123456		
22															
23															

## 读取EXCEL

```
const path = require("path");
const fs = require("fs");
const xlsx = require("node-xlsx");

//parse接收的数据类型是string或ArrayBuffer
//什么是ArrayBuffer我不知道 fs.readFileSync返回的是一个Buff

let result = xlsx.parse(fs.readFileSync(path.join(__dirname,
"./stuinfo.xlsx")));
console.log(result);
```

这个时候得到的结果如下

```
[
  {
    name: 'stuinfo',
    data: [
      [Array], [Array], [Array],
      [Array], [Array], [Array],
      [Array], [Array], [Array],
    ]
  }
]
```

```

        [Array], [Array], [Array],
        [Array], [Array], [Array],
        [Array], [Array], [Array],
        [Array], [Array], [Array],
        [Array]
    ]
},
{ name: 'Sheet1', data: [] }
]

```

它的结果是一个数组，里面的对象中name代表每个excel工作表的名称，data则代表里面的数据

同时我们看到data这个数下面有22项数据，是因为我们读到的第一个excel工作表里面有22行数据，因此我们得出每一行数据都是data里面的一个Array

当我们尝试着去读取里面的某一个工作表的时候，它的格式如下

```

{
  name: 'stuinfo',
  data: [
    [
      'sid',
      'sname',
      'ssex',
      'sbirthday',
      'snation',
      'saddr',
      'smail'
    ],
    [
      2005010101,
      '苏俊丹',
      '女',
      32752,
      '汉族',
      '湖北省/武汉市/武昌区/光谷广场1号',
      '2005010101@softeem.com'
    ],
    [
      2005010102,
      '张小苗',
      '男',
      32753,
      '汉族',
      '湖北省/武汉市/武昌区/光谷广场1号',
      '2005010102@softeem.com'
    ],
    //.....
  ]
}

```

通过分析，我们可以得到data里面的数据是以行为单位，再组成了一个数组【本身是一个二维数组】

现在我们要将上面的数据转换成json格式，怎么办呢

```

[
  {

```

```

        "sid":2005010101,
        "sname":"苏俊丹",
        "ssex":"女",
        "sbirthday":32752,
        "snation":"汉族",
        "saddr":"湖北省/武汉市/武昌区/光谷广场1号",
        "smaill":"2005010102@softeem.com"
    },
    {
        //继续写
    }
]

```

```

let firstExcelTable = result[0];           //得到第1个工作表
//现在我们需要将上面的数据转换成一个json数据，怎么办？
let resultArr = [];
for (let i = 1; i < firstExcelTable.data.length; i++) {
    //这代表每一行数据 都是一个对象
    let obj = {};
    for (let j = 0; j < firstExcelTable.data[i].length; j++) {
        let propertyName = firstExcelTable.data[0][j];           //取属性名
        obj[propertyName] = firstExcelTable.data[i][j];           //对属性名赋值
    }
    resultArr.push(obj);
}
console.log(resultArr);

```

当我们得到了这样的一个JSON数据以后，我们还可以再通过fs模块把它保存为一个json文件，或者后期直接使用一个模板引擎渲染到页面上面

```

fs.writeFileSync(path.join(__dirname, "./biaogege.json"),
JSON.stringify(resultArr), { encoding: "utf-8" });
console.log("写入成功");

```

## 生成EXCEL

在这里，我们演示如果将一个json数据生成一个excel文件

json数据格式如下

```

[
  {
    "instructorid": 1,
    "instructorname": "曾远",
    "instructorsex": "男",
    "instructortel": "1387788998",
    "instructoremail": "www.zenyu123@qq.com",
    "instructorIDCard": "420222199511112524",
    "did": 1
  },
  {
    "instructorid": 2,
    "instructorname": "蒋晓箐",
    "instructorsex": "女",
    "instructortel": "1588884123",
    "instructoremail": "www.jiangjiang123@qq.com",

```

```

        "instructorIDCard": "420222199411112545",
        "did": 2
      }
      //.....省略
    ]
  ]

```

如果要生成一个excel的文件，那么，它的数据格式必须按照你之前读取出来的数据格式，也就是下面的格式

```

[
  {
    name: "Sheet1",
    data: [
      ["instructorid", "instructorname", "instructorsex"],
      [2, "蒋晓箫", "女", "1588884123", "www.jiangjiang123@qq.com"]
      //....省略
    ]
  }
]

```

根据上面的思路，我们先拿到JSON文件里面的数据

### 第一步：读取JSON文件

```

//第一步：先读取json文件
let result1 = fs.readFileSync(path.join(__dirname, "./instructor.json"));
let jsonStr = result1.toString();
let jsonObj = JSON.parse(jsonStr);    //将json字符串转换成对象

```

### 第二步：构造所需要的Excel数据结构

```

//第二步：构造所需要的Excel数据结构
let excelObj = {
  name: "班主任信息",
  data: []
}

```

### 第三步：转换数据

再转换数据的时候，首先先构造第一行的数据，因为第一行的数据就是对象的属性名，也就是excel表头

```

//构造第一行数据，也就是excel的表头，它是我们对象的属性名
let excelFirstLine = Object.keys(jsonObj[0]);
//将第一行的数据添加进去
excelObj.data.push(excelFirstLine);
//接下来添加excel中真正的数据，也就是每个对象的属性值，我们可以使用Object.values()的方法
jsonObj.forEach(item => {
  //Object.values是获取对象的属性值，它与Object.keys()是相对应的
  excelObj.data.push(Object.values(item));
});

```

经过上面三个步骤的转换，我们就已经产生了符合 node-xlsx 的需要的数据格式，最后就可以调用这个模块去产生我们的excel了

#### 第四步：根据数据生成excel文件

```
//buffer可以理解为内在里面的一块数据
let excelBuffer = xlsx.build([excelObj]);
//接下来将内存里面的数据写入硬盘
fs.writeFileSync(path.join(__dirname, "dajiang.xlsx"), excelBuffer, { encoding:
"utf-8" });
console.log("excel文件写入成功");
```

至此为止，我们就已经在本地创建了一个excel文件了