

**移动物联网开发工程实践报告**

**（任务一）**

**题 目 东理教师监考小助手**

**学 院 信工学院**

**专 业** 通信工程

**学 号** 201820130215

**学 生 姓 名**  彭 峰

**指 导 教 师** 饶 志 华

**二○二○ 年 12 月**

1. 需求分析

随着大学规模的增大，高校监考任务的安排越来越多，越来越重要。如何及时、高效的通知监考老师参加监考，给考务工作人员带来不小的压力和负担。考试开始了，学生坐在教室里面，教师还没有到位的事时有发生。以往做法大多为考务工作者在教师工作QQ群里面发一条消息提示监考老师在某某时间和地点有某某科目的考试，但是由于是群发，时有教师没有留意到信息而导致监考缺席。也有个别考务工作者用自己的手机给每位监考老师发送短信提醒教师监考事宜，但是给考务工作者带来巨大的工作量，而且是人为发送，容易导致失误。为响应国家“互联网+”的倡导，致力于减轻教务系统工作人员的压力和减少教师监考迟到和缺席的教学事故的发生，很有必要开发一款及时、高效的自动提醒监考教师有监考任务的教师监考系统。

教师监考系统为我们日后的教务管理方法、思路提供了新方向，符合现代互联网发展的新要求，是互联网+在服务行业的良好示例。

1. 开发任务

程序导入excel格式的“监考数据表”和“教师联系方式表”，自动根据“监考数据表”中的监考时间安排，在特定的时间（如监考前一天及监考前两小时）将“监考数据表”中的监考信息（包含考试时间、考试地点、考试科目、考试班级等信息）以短信的方式发送给监考老师，并获取发送状态报告。

1. 开发过程
   1. 开发环境

操作系统：Win10；

代码编译环境： nodejs ；

编程软件：vscode（任意文本编辑器即可）

Npm依赖包： xlsx ；

短信服务云提供商：云片

* 1. 开发过程

1. 安装nodejs软件，测试查询版本命令测试环境

命令行下输入：Node -v

显示版本号则说明，nodejs环境正常，已经添加到环境变量

1. 阅读并初步学习 nodejs 开发文档
   1. https://www.w3cschool.cn/nodejs/
2. 准备好 apikey 、 模板短信 id 和模板短信对应的接口参数

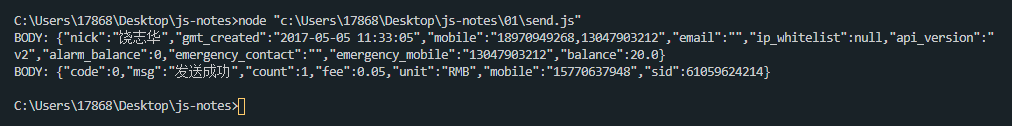
本例参数以 #name#、#code# 为例

1. 新建一个名为send.js 文件：在任意编辑器中编写发送短信代码
   1. 建议使用vscode 编辑器 + nodejs辅助拓展（非必须，只是方便调试）
   2. 注意：本文件中使用 nodejs 默认模块

|  |
| --- |
| // 修改为您的apikey.可在官网（https://www.yunpian.com)登录后获取  var https = require('https'); //通过nodejs载入模块  var qs = require('querystring');  var apikey = 'c403da7e7f1ad6dc28dae19e76acc081';  // 修改为您要发送的手机号码，多个号码用逗号隔开  var mobile = '15770637948';  // 修改为您要发送的短信内容  //var text = '测试使用';  // 指定发送的模板编号  var tpl\_id = 3569052;  // 指定发送模板的内容  var tpl\_value = {'#name#':'彭峰','#code#':'监考地点：软件楼阶5'};  // 查询账户信息https地址  var get\_user\_info\_uri = '/v2/user/get.json';  // 智能匹配模板发送https地址  var sms\_host = 'sms.yunpian.com';  //var voice\_host = 'voice.yunpian.com';  send\_sms\_uri = '/v2/sms/single\_send.json';  // 指定模板发送接口https地址  send\_tpl\_sms\_uri = '/v2/sms/tpl\_single\_send.json';  //-------------调用函数部分-----------------------  query\_user\_info(get\_user\_info\_uri,apikey);  //send\_sms(send\_sms\_uri,apikey,mobile,text);  send\_tpl\_sms(send\_tpl\_sms\_uri,apikey,mobile,tpl\_id,tpl\_value);  //send\_voice\_sms(send\_voice\_uri,apikey,mobile,code);  //------------调用函数部分 END ---------------------  //-----------定义各个函数部分 ------------------  function query\_user\_info(uri,apikey){  var post\_data = {  'apikey': apikey,  };//这是需要提交的数据  var content = qs.stringify(post\_data);  post(uri,content,sms\_host);  }  /\*  function send\_sms(uri,apikey,mobile,text){  var post\_data = {  'apikey': apikey,  'mobile':mobile,  'text':text,  };//这是需要提交的数据  var content = qs.stringify(post\_data);  post(uri,content,sms\_host);  }\*/  function send\_tpl\_sms(uri,apikey,mobile,tpl\_id,tpl\_value){  var post\_data = {  'apikey': apikey,  'mobile':mobile,  'tpl\_id':tpl\_id,  'tpl\_value':qs.stringify(tpl\_value),  };//这是需要提交的数据  var content = qs.stringify(post\_data);  post(uri,content,sms\_host);  }/\*  function send\_voice\_sms(uri,apikey,mobile,code){  var post\_data = {  'apikey': apikey,  'mobile':mobile,  'code':code,  };//这是需要提交的数据  var content = qs.stringify(post\_data);  console.log(content);  post(uri,content,voice\_host);  }\*/  function post(uri,content,host){  var options = {  hostname: host,  port: 443,  path: uri,  method: 'POST',  headers: {  'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8'  }  };  var req = https.request(options, function (res) {  // console.log('STATUS: ' + res.statusCode);  // console.log('HEADERS: ' + JSON.stringify(res.headers));  res.setEncoding('utf8');  res.on('data', function (chunk) {  console.log('BODY: ' + chunk);  });  });  //console.log(content);  req.write(content);  req.end();  }  //-----------定义各个函数部分 END ------------------ |

运行方式：

1. vscode 运行：最好安装一个插件：coderunner，直接按下F5 选择 nodejs 编译就可以了
2. 命令行下运行：



1. 第一个BODY 表示查询当前的用户信息：姓名昵称、账户创建时间、手机号、邮箱、ip白名单、api版本、余额警报、紧急联系人电话、余额

2）第二个BODY表示当前使用模板发送单条短信，返回内容为代码0、消息发送状态、发送条数、本次发送费用、计费方式、发送目的手机号、本次短信发送id

1. 新建一个名为read.js文件：编写处理excel表格中的数据

此步骤需要使用 npm 外部依赖包来读写 excel 文件，优先考虑使用 js-xlsx 依赖包

具体操作参考 <https://www.cnblogs.com/unreal-feather/archive/2004/01/13/12794129.html>

注意事项：

1. 需要修改 npm 镜像源 ，加速下载依赖包
2. 依赖包最好下载到代码文件同目录文件夹下
3. 如需使用以下代码实现，请修改 文件的相对路径
4. 各行代码实现的功能见注释

实现excel数据处理的代码如下：

|  |
| --- |
| var XLSX =require('xlsx')  var fs=require('fs')  const name\_phone = XLSX.readFile('./01/监考老师电话号码库.xlsx')  const ws1 = name\_phone.Sheets[name\_phone.SheetNames[0]]  const data1 = XLSX.utils.sheet\_to\_json(ws1)  fs.writeFile('./01/phone\_number.json', JSON.stringify(data1, null,8),function (err) {  if (err){  console.log('写入失败')  }  else{  console.log('电话信息文件写入成功,请查看 phone\_number.json 文件')  }  })  //console.log(data1)  // https://www.cnblogs.com/unreal-feather/archive/2004/01/13/12794129.html  const exam = XLSX.readFile('./01/14-17周专业课考试安排表L.xlsx')  const ws2 = exam.Sheets[exam.SheetNames[0]]  //使用！ref设置表格的有效范围 A2：G13  ws2['!ref']= XLSX.utils.encode\_range({s:{c:0, r:1}, e:{c:6, r:12}})  const data2 = XLSX.utils.sheet\_to\_json(ws2) //将表格解析成json格式  console.log("查询监考老师电话：")  for(var i=0;i<data2.length;i++)  {  // 补全 每一条监考信息，如果某个位置为空，则用上一条信息同位置的信息代替  if(data2[i].考试课程 ==null)  data2[i].考试课程=data2[i-1].考试课程.trim()  if(data2[i].考试时间 ==null)  data2[i].考试时间=data2[i-1].考试时间  if(data2[i].任课教师 ==null)  data2[i].任课教师=data2[i-1].任课教师  //用于删除字符中开头和结尾的空格  data2[i].考试课程=data2[i].考试课程.trim()  //以顿号为标志分割监考教师姓名  data2[i].监考=data2[i].监考.split("、")  // console.log(data2[i])  var name2phone  //链接两个表查询对应监考老师的电话号码  for(var j=0;j<data1.length;j++)  {  if(data2[i].监考[0]==data1[j].姓名)  name2phone = data2[i].监考[0] +" "+ data1[j].电话    }    console.log(name2phone)  }  fs.writeFile('./01/exam.json', JSON.stringify(data2, null,8),function (err) {  if (err){  console.log('写入失败')  }  else{  console.log('监考信息文件写入成功,请查看 exam.json 文件')  }  }) |

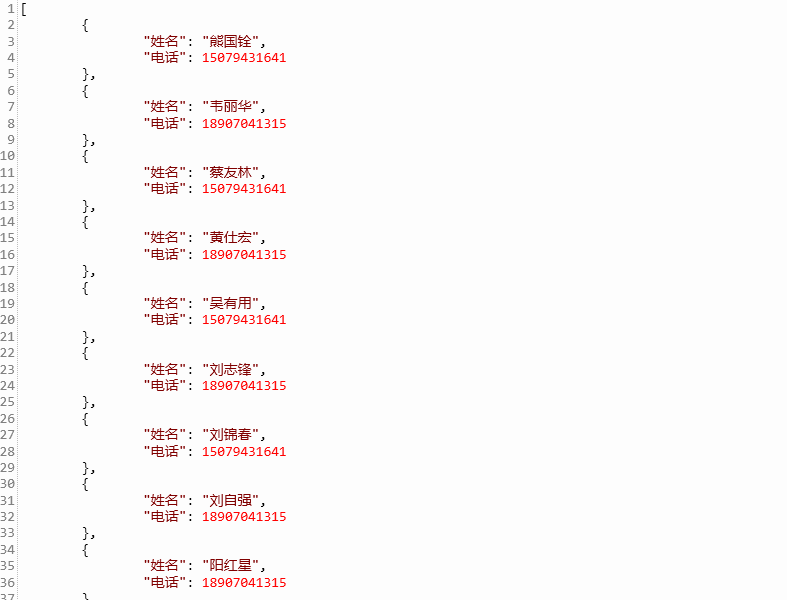
运行结果会自动在同级目录下生成两个 json 文件，分别为：exam.json 和 phone\_number.json

包含了 监考信息 和 各位老师的电话号码信息，文件内容均为 json 格式，运行结果图片如下



如何处理数据以及解决相关问题，已经在代码注释中详细说明



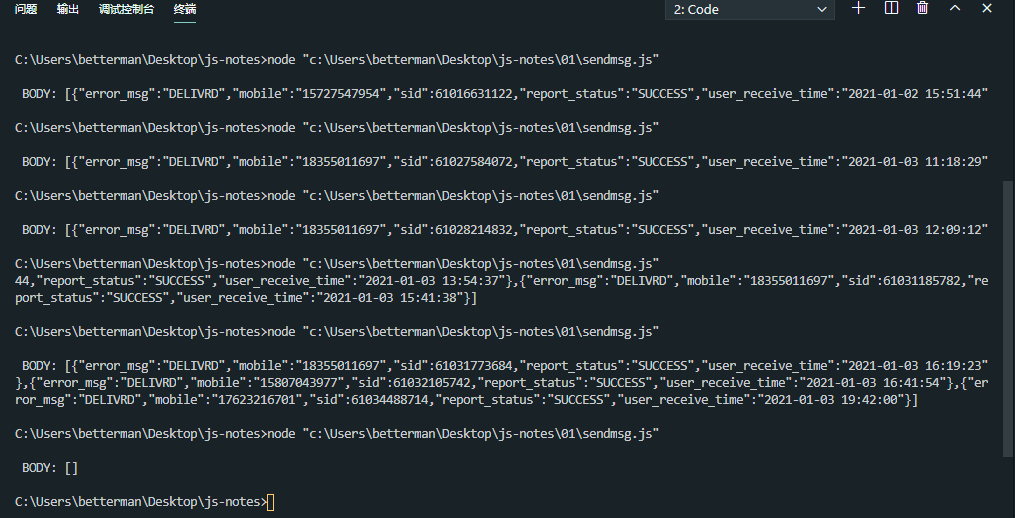


1. 查询已发送短信的接口，并保存记录

创建一个名为 countSMS.js的文件，编辑以下查询代码

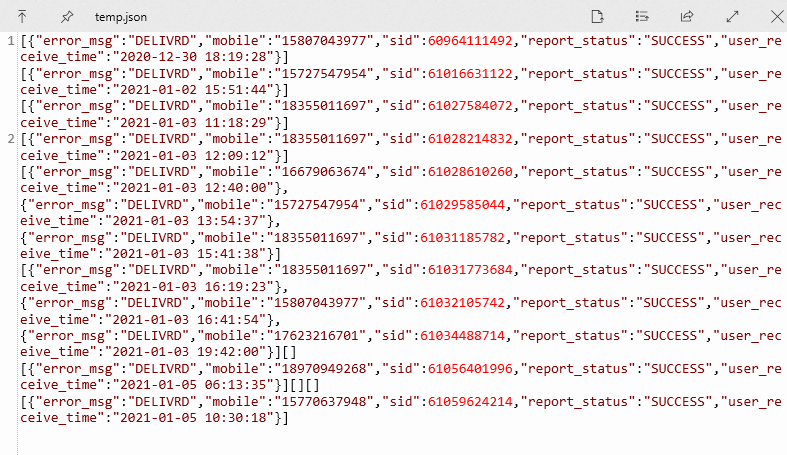
|  |
| --- |
| // 修改为您的apikey.可在官网（https://www.yunpian.com)登录后获取  var https = require('https');  var fs = require("fs")  var os = require("os");  var path = require("path");  var qs = require('querystring');  var apikey = 'c403da7e7f1ad6dc28dae19e76acc081';  var tpl\_value ={'#page\_size#':5}  // 查询账户信息https地址  var get\_user\_info\_uri = '/v2/user/get.json';  // 智能匹配模板发送https地址  var sms\_host = 'sms.yunpian.com';  var page\_size = 5;  send\_tpl\_sms\_uri = '/v2/sms/pull\_status.json';  send\_tpl\_sms(send\_tpl\_sms\_uri,apikey,sms\_host,page\_size);  function send\_tpl\_sms(uri,apikey,sms\_host,page\_size){  var post\_data = {  'apikey': apikey,  'page\_size':page\_size,  // 'tpl\_value':qs.stringify(tpl\_value),  };//这是需要提交的数据  var content = qs.stringify(post\_data);  post(uri,content,sms\_host);  }  function post(uri,content,host){  var options = {  hostname: host,  port: 443,  path: uri,  method: 'POST',  headers: {  'Accept':'application/json;charset=utf-8',  'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8'  }  };  var req = https.request(options, function (res) {  // console.log('STATUS: ' + res.statusCode);  // console.log('HEADERS: ' + JSON.stringify(res.headers));  res.setEncoding('utf8');  res.on('data', function (chunk) {  console.log('\n BODY: ' + chunk );  //将每次查询的结果作为一个数组附加到 文件末尾  fs.appendFileSync('./01/temp.json', chunk, { mode: 0o600 });  });  });  //console.log(content);  req.write(content);  req.end();  } |

运行结果如下：



此程序会将查询结果保存到同目录下的 temp.json 文件中，每次查询的结果会附加到文件末尾

文件内容如下：



此步骤，如需继续进行数据处理——统计总信息条数，筛选短信id等，需要学习 json 文件格式以及如何使用nodejs 增删查改 json 文件，其实这些已经在 读取并处理 excel 表格中的数据时已经用到了这些方法，请仔细查看 read.js 文件代码，代码简短易读，风格类似 C++和matlab

关于json 文件的简单说明：

1. 学会 区分 {} 和 [ ] 的作用
2. 学会引用对象和数组
3. 学会对象和数组的嵌套及其使用方法
4. 开发遇到的关键问题及解决方法



图 2 多行监考教师和一行监考科目对应关系

问题产生原因：

解决办法：

问题0、npm下载依赖包速度慢

1. 发送短信在手机上显示乱码原因
2. 发送域名和发送路径拼接的问题
3. 一个是处理合并单元格的问题，有的行读取为空。
4. 处理监考老师姓名分割的问题
5. 匹配监考教师和电话号码
6. 处理考试时间分割的问题
7. 根据考试时间定时发送短信
8. 心得和体会
9. 本项目已经实现excel表格的数据处理、发送短信、查询用户信息和查询发送记录等功能。

2、可继续使用 electron 实现可视化界面，用于展示文件数据，手动选择监考教师发送短信，也可以使用 nodejs 搭建本地服务器 构建一个网页 使用，只需要将excel 文件处理成json 格式文件，嵌入到应用中即可，如需分发给用户使用，可使用 electron 模块将应用打包成exe文件，无需nodejs 环境也可以使用。

1. 尚未实现的功能：
2. 根据考试时间定时发送短信：可设置定间触发事件，回调发送短信函数来发送短信，分割日期后，找到年月日，再根据上午下午以及课程来设置时间
3. 考试时间分割问题：类似分割监考老师姓名
4. 关于乱码的问题：JavaScript 语言自身只有字符串数据类型，没有二进制数据类型。如需使用二进制数据类型，需用使用 buffer
5. 本项目使用nodejs 在数据处理上极具优势，json 文件格式巧妙地避免了乱码问题，并且nodejs 并发性能高，可异步，可多线程，可添加回调函数。除此之外，js文件封装处理的较好，可以理解为每一个模块都是很多个js文件，载入一个模块，相当于调用了这个模块的接口，极大地提高了可移植性。
6. 心得：

通过本次课程设计，我熟悉了个人小项目的基本开发流程，能够熟练查询开发文档来解决项目中的一些问题，将理论与实际相结合，提高了自己的解决问题的能力。遇到问题，我可能会做许多尝试，但失败的尝试并不意味着没有尽头，有时候我们还需要停下来思考解决问题的其他方法，说不定就成功了。

我深知在开发过程中我还有许多不足，不仅仅是理论知识方面的不足，还有团队协作方面的不足，一个人的力量总是渺小的，一个人的眼界总是狭隘的，我相信团队协作，共同开发，才能够在实际项目中，保持进度，保证高效！