



链滴

使用 node 实现社区自动签到

作者: [csfwff](#)

原文链接: <https://ld246.com/article/1566187415790>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



身为一个菜鸡，也没啥好水的，就写下自动签到吧

这个签到程序，跟着社区升级，更新了好几版，从最开始只要带着cookie请求签到页面就行，到后来要先获取token，再到现在的需要更新cookie，中途踩了几个小坑，随便看看吧

1. 需求分析

自动签到嘛，只要到了指定时间带着身份信息去请求指定的页面或接口就行，最后以防万一，把签到果发个邮件给自己的邮箱，好了，分析完毕

2. 初始化项目

首先，安装node，已安装，跳过。

然后PowerShell执行`yarn init`，输入名称，描述啥的，还有入口文件，爱写啥写啥写，我写`app.js`，后在项目文件夹下建立入口文件，也就是`app.js`，初始化完成。

3. 定时任务

本着能用别人的绝不自己写的原则，先安装定时任务模块

继续用yarn安装node-schedule: `yarn add node-schedule`

然后在app.js中引入

```
const schedule = require('node-schedule');
const signTask = () => {
  schedule.scheduleJob('2 0 0 * * *', () => {
    let nowTime = new Date();
    console.log("\n\n\n")
    console.log("----->" + nowTime.toLocaleDateString() + "-开始签到<-----");
    hacpaiGetSignUrl(); //这里去签到，后续会说明
```

```
    })  
  }  
  signTask(); //启动定时任务
```

其中`scheduleJob`的第一个参数是`* * * * *`，6个星号，从左往右分别是秒分时日月年，因为我定的是每天0点0分2秒开始签到，所以写`2 0 0 * * *`，如果是每小时的第25分15秒执行，那么写`15 25 * * *`，可以设定每分钟执行一次测试定时任务有没有正常启动

4. 获取cookie

由于我懒得写登录，就直接F12拿cookie了，只需要`JSESSIONID`和`symphony`两个部分，搞一个数存放，0放`JSESSIONID`，1放`symphony`，最后`JSON.stringify(cookieArray)`一下转字符串写入项目文件夹下的`cookie.txt`中，大概长这样

```
["JSESSIONID=lalalala换成你自己的;Path=/", "symphony=lalalala换成你自己的;Path=/;Expires=un, 25-Aug-2019 16:00:03 GMT;Max-Age=604800;Secure;HttpOnly"]
```

5. 获取签到链接

安装请求模块: `yarn add request`

安装cheerio用于解析网页: `yarn add cheerio`

在`app.js`中加入请求部分，具体过程看注释

```
const request = require('request');  
const cheerio = require('cheerio');  
const fs = require('fs'); //用于读写文件  
  
const hacpaiGetSignUrl = () => {  
  //先读取cookie, 需要转成utf-8  
  let hacCookie = fs.readFileSync('cookie.txt').toString('utf-8');  
  hacCookie = JSON.parse(hacCookie)  
  let url = 'https://hacpai.com/activity/checkin'  
  //设定请求参数  
  let options = {  
    url: url,  
    headers: {  
      'Upgrade-Insecure-Requests': 1,  
      'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36',  
      'cookie': hacCookie  
    }  
  }  
  let nowTime = new Date();  
  //发起请求  
  request(options, (err, res, body) => {  
    if (err) {  
      console.log(nowTime.toLocaleTimeString() + "---->签到获取地址失败---->\n" + err);  
      sendMailTast("黑客派签到地址请求失败", "") //请求失败时直接发邮件  
    } else {  
      console.log(nowTime.toLocaleTimeString() + "---->签到获取地址成功---->\n");  
      //如果返回头里包含set-cookie字段, 需要更新cookie  
      if (res.headers['set-cookie'] != undefined) {
```

```

        //更新cookie
        writeCookie(res.headers['set-cookie'], hacCookie)
    }
    //开始解析网页
    let $ = cheerio.load(body)
    let signUrl = ""
    try {
        signUrl = $('btn.green').get(0).attribs.href;
        console.log("---->签到地址" + signUrl + "\n");
        hacpaiSignRequest(signUrl) //获取地址成功, 去请求即可签到成功
    } catch (e) {
        signUrl = '签到地址异常'
        sendMailTast("黑客派签到地址异常", "") //解析异常的时候直接发邮件
    }
}
})
}

```

其中, 请求的时候需要带上UA, 否则会报Too Many Requests

更新cookie封个方法, 如下

```

const writeCookie = (setCookie, oldCookie) => {
    setCookie.forEach((e) => {
        if (e.indexOf('JSESSIONID') !== -1) {
            oldCookie[0] = e
        }
        if (e.indexOf('symphony') !== -1) {
            oldCookie[1] = e
        }
    })
    fs.writeFileSync('cookie.txt', JSON.stringify(oldCookie))
}

```

6. 签到

拿到签到地址后直接请求就可以了, 方法跟获取地址一样

```

//将请求地址作为参数传过来
const hacpaiSignRequest = (signUrl) => {
    //读cookie
    let hacCookie = fs.readFileSync('cookie.txt').toString('utf-8');
    hacCookie = JSON.parse(hacCookie)
    //设置请求参数
    let options = {
        url: signUrl,
        headers: {
            'Referer': 'https://hacpai.com/activity/checkin',
            'Upgrade-Insecure-Requests': 1,
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36',
            'cookie': hacCookie
        }
    }
}

```

```

let nowTime = new Date();
console.log("时间: " + nowTime.toLocaleTimeString());
//发起请求
request(options, (err, res, body) => {
  nowtime = new Date();
  if (err) {
    console.log(nowTime.toLocaleTimeString() + "---->签到请求失败---->\n" + err);
    //请求失败发邮件
    sendMailTast("黑客派签到结果", nowTime.toLocaleTimeString() + "---->签到请求失败--
->\n" + err);
  } else {
    console.log(nowTime.toLocaleTimeString() + "---->签到请求成功---->\n");
    if (res.headers['set-cookie'] != undefined) {
      writeCookie(res.headers['set-cookie'], hacCookie)
    }
    //解析网页获取今天签到获得的积分
    let $ = cheerio.load(body)
    let num = "0"
    try {
      num = $('code').get(0).children[0].data;
    } catch (e) {
      num = '签到异常'
    }
    //签到完成, 发送邮件
    sendMailTast("黑客派签到结果" + num, "获得积分: " + num + " \n" + nowTime.to
ocaleTimeString() + "---->签到请求成功---->\n" + "status---->" + res.statusCode + "\n
到获得积分-->" + num);
  }
})
}

```

7. 发送邮件

安装发送邮件的模块 `yarn add nodemailer`

去自己的邮箱, 开启POP3和SMTP服务, 理论上是会让设一个授权码什么的, 忘记了
然后, 发邮件, 我用的163, 仅供参考

```

const nodemailer = require('nodemailer');

const sendMailTast = (title, content) => {
  let transporter = nodemailer.createTransport({
    host: 'smtp.163.com',
    port: 465,
    secure: true,
    auth: {
      user: '你的用户名',
      pass: '授权码或密码, 忘记了, 理论上是要授权码, 反正试试就行'
    }
  });
  //邮件信息
  let mailOptions = {
    from: '"签到"<发件邮箱>',
    to: '接收邮箱',

```

```

    subject: title,
    text: content
  };
  //发送邮件
  transporter.sendMail(mailOptions, (error, info) => {
    let nowTime = new Date();
    if (error) {
      //发送异常
      console.log("时间: " + nowTime.toLocaleTimeString() + title + "-->邮件发送失败-->\n" + error);
    } else {
      //发送成功
      console.log("时间: " + nowTime.toLocaleTimeString() + title + "-->邮件发送成功");
    }
    //邮件发送完毕, 今日签到完成
    console.log("----->" + nowTime.toLocaleDateString() + title + "-签到结束<-----");
  })
}

```

邮箱的host和port可以去邮箱设置里找, SMTP默认非SSL为25端口, 但是一般在服务器上这个端口被禁的, 所以改用SSL的465端口

8. 丢服务器跑起来

把app.js整理一下, 把整个文件夹丢服务端, node_modules不用放, 在服务端yarn install一下就可以了, 最后node app启动就ok了, 接下就只要每天早上看下邮件有没有正常签到就可以了

9. EOF

没了, 后会有期