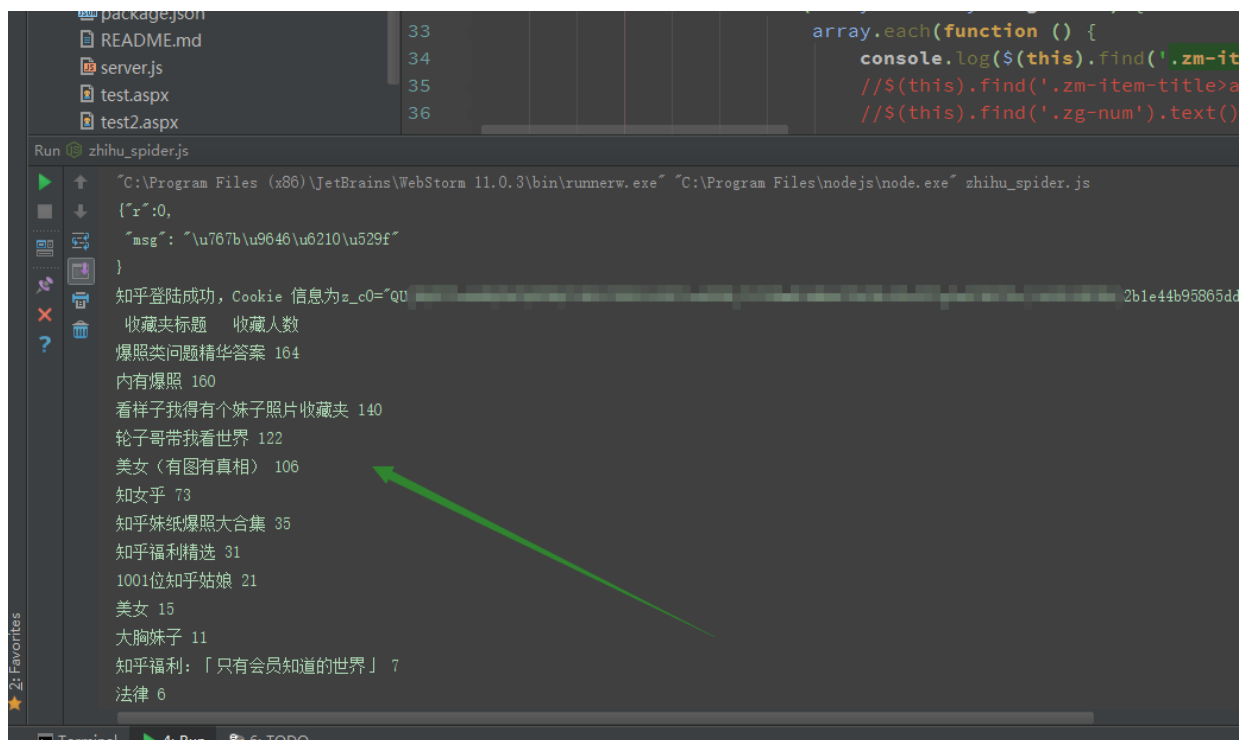


## Node.js：实现知乎(www.zhihu.com)模拟登陆，获取用户关注主题

Lumia1020 2016-03-28 2851 查看

+添加收藏



前一段时间，在瞎看看 Node.js，便研究通过 Node.js 实现知乎模拟登陆。相信，有很多网站有登陆权限设置，如若用户未登陆，将会跳转至首页提醒用户登陆，无法浏览部分页面。

如若是 b/s 架构，肯定是离不开 http(s) 协议，而 http(s) 协议又为无状态，为了实现状态保存，出现了 cookie/session，因此，登陆后，保存用户登陆状态，无非不就是利用 cookie/session 实现以上功能。cookie 与 session 有所不同，其中 cookie 保存在客户端，而 session 则保存在服务器端，不过，session 其中的 sessionid 将会保存中 cookie 中。

终上所述，实现模拟登陆，获取用户关注主题，将分为以下几步。

首先，进行模拟登陆，不过，在登陆过程中，要将提交至服务器参数填充完整(一般包括用户名、密码、随机数等)。

在登陆成功后，服务器将会通过 Set-Cookie 指令向客户端写入 Cookie。之后，一般网站即会进行跳转至登陆前页面


(所以，需要在页面跳转前获取 Cookie 值进行保存)。

向目标 URL 发送请求，此时，该目标 URL 在请求报文头中的 cookie 信息加入前一步获取

的登陆 cookie 信息。

而实现以上功能，肯定离不开 Chrome F12 开发者工具分析，而在 Node.js 中，则利用到了 Superagent、Cheerio 模块，其中 Superagent 可模拟浏览器请求，而 Cheerio 模块则类似于服务器端 jQuery，经常 DOM 操作增删查改的同学，肯定不会陌生，同时，其 API 也与浏览器端的 jQuery 类库极其相似。


一 首先，分析权限页面所需 Cookie 信息，打开 Chrome F12 开发者工具，选择 Sources -- Cookies，可查看页面 Cookie 信息，可点击 Cookie 值进行删除操作，后再刷新页面，看是否会跳转至登陆页面。通过重重排除可知，该页面需要名为 z\_co 的 cookie 信息，方可打开目标权限页面，从而，在模拟登陆，保存 set-cookie 指令中名为 z\_c0 的 cookie 信息。




1113

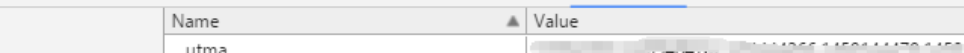
## 超级英雄的出现，让他们世界变好了吗？

方十一，组织研究决定由你来扯这个犊子


[▼](#) 【丹佛枪击案故事：13岁女孩挺身击退8岁女孩】美国射击锦标赛丹佛枪击赛开幕式枪击案发生时，13岁女孩凯兰也在场，她当时努力拯救凯兰的9岁女孩，但凯兰还是不幸遇难。凯兰被枪击倒的瞬间，她身上7处致命伤，按顺序，当时挺身而出救助他人的青少年不止凯兰一个。



内有较多图片，大多是漫画，不建议手机阅读，若是手机党，请确保您处于WIFI状态或者不在乎流量。看到这个题目确实有很多想说的.....因为我作为半吊子美漫迷，知道的一点可怜巴巴的东西都是关于英雄们的，所以目前上也是用拍图软件，上边这图



二 接下来，同样采用 F12 开发者工具分析，在登陆过程，Post 请求中传递的参数信息，包括 `_xsrf`、`email`、`password`、`remeber_me` 四个参数，除 `_xsrf` 外，相当于都为已知参数。再通过 `ctrl+shift+f` 全局搜索 `_xsrf` 可知，其值为一个隐藏 `input type='hidden'` 标签 `value` 属性，对 Dom 参数及值进行修改，因而，可进一步分析此参数，可否随意填写。

此处在登录请求 url

通过登陆成功响应报文头分析可知，服务器通过 Set-Cookie 指令向浏览器写入了三个 Cookie 信息，其中分别为 n\_c、unlock\_ticket、z\_c0，而联系第一步分析，保存 z\_c0 cookie 即可

Form Data：其中 \_xsrf 为页面中信息，password 为用户登陆密码，email 为用户登陆帐号，remember\_me 则标记是否记住用户密码

Headers

Content-Type: application/json  
Date: Mon, 28 Mar 2016 06:54:30 GMT  
Pragma: no-cache  
Server: nnws/1.7.3.6  
Set-Cookie: n\_c=...  
Set-Cookie: unlock\_ticket=...  
Set-Cookie: z\_c0=...  
Vary: Accept-Encoding  
X-Frame-Options: DENY  
X-Req-ID: 1C0339D856F8D513  
X-Za-Response-Id: 834c31fb852f4a68a6be32c8b8efcfe5

Request Headers

Accept: /\*/\*  
Accept-Encoding: gzip, deflate  
Accept-Language: zh-CN,zh;q=0.8  
Cache-Control: no-cache  
Connection: keep-alive  
Content-Length: 105  
Content-Type: application/x-www-form-urlencoded; charset=UTF-8  
Cookie: ...  
Host: www.zhihu.com  
Origin: https://www.zhihu.com  
Pragma: no-cache  
Referer: https://www.zhihu.com/  
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML like Gecko) Chrome/48.0.2564.101 Safari/537.36

X-Requested-With: XMLHttpRequest

Form Data

\_xsrf: ...  
password: ...  
remember\_me: true  
email: ...

通过，对于登陆页面 Dom 结构分析，可明显发现 id 为 \_xsrf 标签信息，而后，本人尝试修改其 value 值，而后点击提交，发现其在 JS 中应该有做相关处理，而并非本人修改后的值。而该值获取，亦可采取最为简捷的 Superagent 模块获取页面全部字符串，之后，即可采用 cheerio 获取 id 为 \_xsrf 标签，从而获取其值。

```
<!DOCTYPE html>
<html lang="zh-CN" class="show-app-promotion-bar cssanimations csstransforms csstransitions flexbox no-touchevents no-mobile">
  <#shadow-root
    <style type="text/css" id="1054153606806"></style>
    <head>...</head>
    <body class="zhi no-auth">
      <div class="index-main">
        ::before
        <div class="index-main-body">
          <div class="index-header">...</div>
          <div class="desk-front sign-flow clearfix sign-flow-simple">
            ::before
            <div class="index-tab-navs">...</div>
            <div class="view view-signin" style="display: block;">
              <form novalidate="novalidate">
                <input type="hidden" name="_xsrft" value="fe57ebb5dfcaf4587c19e8a45d7f779f">...</input>
                <div class="group-inputs">...</div>
                <div class="failure" id="summary">...</div>
                <div class="button-wrapper command">...</div>
                <div class="signin-misc-wrapper clearfix">...</div>
                <div class="weibo-signup-wrapper">...</div>
              </form>
            </div>
            <div class="view view-signup selected" style="display: none;">
              <form class="zu-side-login-box" action="/register/email" id="sign-form-1" autocomplete="off" novalidate="novalidate">
                <input type="password" hidden>...</input>
                <input type="hidden" name="_xsrft" value="fe57ebb5dfcaf4587c19e8a45d7f779f">...</input>
                <div class="group-inputs">...</div>
                <div class="failure" id="summary">...</div>
                <div class="button-wrapper command">...</div>
                <div class="weibo-signup-wrapper">...</div>
              </form>
            </div>
            ::after
          </div>
        </div>
      </div>
    </div>
  </div>
</html>
```

html body.zhi.no-auth div.index-main div.index-main-body div.desk-front.sign-flow.clearfix.sign-flow-simple div.view.view-signin form input

\_xsrft 2 of 3 Cancel

```
// 引入 superagent、cheerio
var superagent= require("superagent");
var cheerio=require("cheerio");

// 登陆 url、目标 url
var url={
  url:"http://www.zhihu.com/",
  login_url:"http://www.zhihu.com/login/email",
  target_url:"https://www.zhihu.com/collections"
};

// 浏览器请求报文头部部分信息
var browserMsg={
  "User-Agent":"Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.116 Safari/537.36",
  'Content-Type':'application/x-www-form-urlencoded; charset=UTF-8'
};

var cookie;

// post 参数信息，其中，还差先前分析的 _xsrft 信息
var loginMsg=
{
  password:"xxxx ",
  remember_me:true,
```

```

    email:"xxxxx"
};

// 获取 _xrsf 值
function getXrsf(){

    superagent.get(url.url).end(function(err,res){
        if(!err){
            var $=cheerio.load(res.text);
            loginMsg._xsrp=$('[name=_xsrp]').attr('value');
        }else
            console.dir(err);

    });
}

// 发送登陆请求，获取 cookie 信息
function getLoginCookie() {
    // 首先，需在 set 方法中设置请求报文中参数，以性器官免服务器端有针对非浏览器请求做相关处理
    // send 方法中设置 post 请求中需提交的参数
    // redirects 方法调用，其中参数为 0，为了避免在用户登陆成功后，引起的页面重新刷新，从而无法获取 cookie
    superagent.post(url.login_url).set(browserMsg).send(loginMsg).redirects(0).end(function (err, response) {
        if (!err) {
            cookie = response.headers["set-cookie"];
            console.dir(cookie);
        } else

            console.dir(err);

    });
}

// 根据 cookie，获取 target 页面关注信息
// 通过分析可知，仅取出 z_c0 的 cookie 即可，而 getLoginCookie 方法返回为一个 cookie 数组，稍做处理即可
function getFollower(){
    superagent.get(url.target_url).set("Cookie",cookie).set(browserMsg).end(function(err,response){
        if (err) {
            console.log(err);
        } else {

            var $ = cheerio.load(response.text);

            // 此处，同样利用 F12 开发者工具，分析页面 Dom 结构，利用 cheerio 模块匹配元素
            var array = $('#zh-favlist-following-wrap .zm-item');
            console.log("收藏夹标题 " + " " + " 收藏人数");
            if (array && array.length > 0) {
                array.each(function () {
                    console.log($(this).find('.zm-item-title>a').text() + " " + $(this).find('.zg-num').text() ? $(

```

```
this).find('.zg-num').text() : "0"));
    //$(this).find('.zm-item-title>a').text();
    //$(this).find('.zg-num').text();

    });
}

}
});

}
```

最近，身体一天天比一天差了，楼主营养跟不上了。（Node.js 新手，代码太渣，勿喷！！  
！）关于 Superagent、cheerio 模块使用可查看 api 文档。

参考资料：

<https://cnodejs.org/topic/5203a71844e76d216a727d2e>

<https://cnodejs.org/topic/5378720ed6e2d16149fa16bd>

🔖 标签：

## 免费送书活动

- |                               |                                     |
|-------------------------------|-------------------------------------|
| — 《智能时代:大数据与智能革命重新定义未来》       | — 《O'Reilly精品图书系列:JavaScript权威指南(第6 |
| — 《C++程序设计语言(第4部分:标准库)(原书第4   | — 《吴军博士 浪潮之巅(第3版)(套装共2册)》           |
| — 《计算机科学丛书:Php和Mysql Web开发(原书 | — 《图灵程序设计丛书:Python基础教程(第2版)(修       |
| — 《数据结构与算法分析:C++语言描述(第2版)》    | — 《世界著名计算机教材精选:C++面向对象程序            |
| — 《PHP从入门到精通(第3版)(附光盘)》       | — 《C++ Primer(中文版)(第5版)》            |
| — 《计算机科学丛书:Java语言程序设计(基础篇)(   | — 《Linux命令行与shell脚本编程大全(第2版)》       |
| — 《Linux命令行与shell脚本编程大全 第3版》  | — 《深入理解Nginx:模块开发与架构解析(第2版)》        |
| — 《HTTP权威指南》                  | — 《高性能MySQL(第3版)》                   |
| — 《细说PHP(第2版)(附DVD光盘1张)》      | — 《PHP开发实战1200例(第1卷)(附DVD-ROM光盘     |

— [《深入PHP:面向对象、模式与实践\(第3版\)》](#)

— [《Android 5.0开发范例代码大全\(第4版\)》](#)

## 相关文章推荐

— [使用jQuery和CSS3制作数字时钟\(CSS3篇\)](#)

— [纯CSS技术实现链接列表悬停效果](#)

— [jQuery+PHP动态数字展示效果](#)

— [jQuery插件实现的加载图片和页面效果](#)

— [jQuery在线选座订座（影院篇）](#)

— [HTML5数字时钟之闹钟](#)

— [使用moment.js轻松管理日期和时间](#)

— [Flexslider图片轮播、文字图片相结合滑动切换](#)

— [使用CSS3和jQuery可伸缩的搜索条](#)

— [jQuery插件实现的页面功能介绍引导页效果](#)

— [jQuery数字加减插件](#)

— [jQuery在线选座（高铁版）](#)

— [PHP+jQuery+Ajax多图片上传](#)

— [使用jQuery和CSS3制作数字时钟\(jQuery篇\)](#)

— [页面右侧浮动固定层返回顶部按钮带二维码展](#)

— [jQuery圆形统计图（百分比）](#)