

# Python爬虫爬取知乎小结

晴明 马哥Linux运维 2 days ago

每日一个Linux、Python干货



关注的人都加薪了

最近学习了一点网络爬虫，并实现了使用Python来爬取知乎的一些功能，这里做一个小的总结。网络爬虫是指通过一定的规则自动的从网上抓取一些信息的程序或脚本。我们知道机器学习和数据挖掘等都是从大量的数据出发，找到一些有价值有规律的东西，而爬虫则可以帮助我们解决获取数据难的问题，因此网络爬虫是我们应该掌握的一个技巧。

Python有很多开源工具包供我们使用，我这里使用了requests、BeautifulSoup4、json等包。requests模块帮助我们实现http请求，bs4模块和json模块帮助我们从获取到的数据中提取一些想要的信息，几个模块的具体功能这里不具体展开。下面我分功能来介绍如何爬取知乎。

## 模拟登录

要想实现对知乎的爬取，首先我们要实现模拟登录，因为不登录的话好多信息我们都无法访问。下面是登录函数，这里我直接使用了知乎用户fireling的登录函数，具体如下。其中你要在函数中的data里填上你的登录账号和密码，然后在爬虫之前先执行这个函数，不出意外的话你就登录成功了，这时你就可以继续抓取想要的数据。注意，在首次使用该函数时，程序会要求你手动输入captcha码，输入之后当前文件夹会多出cookiefile文件和zhihucaptcha.gif，前者保留了cookie信息，后者则保存了验证码，之后再去模拟登录时，程序会自动帮我们填上验证码。

```
def login():
    url = 'http://www.zhihu.com'
    loginURL = 'http://www.zhihu.com/login/email'
```

```
headers = {
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:41.0) Gecko/20100101 Firefox/41.0",
    "Referer": "http://www.zhihu.com/",
    "Host": "www.zhihu.com",
}

data = {
    'email': 'you@example.com',
    'password': '*****',
    'rememberme': "true",
}

global s
s = requests.session()
global xsrf
if os.path.exists('cookiefile'):
    with open('cookiefile') as f:
        cookie = json.load(f)
    s.cookies.update(cookie)
req1 = s.get(url, headers=headers)
soup = BeautifulSoup(req1.text, "html.parser")
xsrf = soup.find('input', {'name': '_xsrf', 'type': 'hidden'}).get('value')
# 建立一个zhihu.html文件,用于验证是否登陆成功
with open('zhihu.html', 'w') as f:
    f.write(req1.content)
else:
    req = s.get(url, headers=headers)
    print req

soup = BeautifulSoup(req.text, "html.parser")
xsrf = soup.find('input', {'name': '_xsrf', 'type': 'hidden'}).get('value')

data['_xsrf'] = xsrf

timestamp = int(time.time() * 1000)
captchaURL = 'http://www.zhihu.com/captcha.gif?' + str(timestamp)
print captchaURL

with open('zhihucaptcha.gif', 'wb') as f:
    captchaREQ = s.get(captchaURL, headers=headers)
    f.write(captchaREQ.content)
loginCaptcha = raw_input("input captcha:\n").strip()
data['captcha'] = loginCaptcha
print data
loginREQ = s.post(loginURL, headers=headers, data=data)
if not loginREQ.json()['r']:
    print s.cookies.get_dict()
    with open('cookiefile', 'wb') as f:
        json.dump(s.cookies.get_dict(), f)
else:
    print 'login fail'
```

需要注意的是，在login函数中有一个全局变量s=reequests.session()，我们用这个全局变量来访问知乎，整个爬取过程中，该对象都会保持我们的持续模拟登录。

## 获取用户基本信息

知乎上每个用户都有一个唯一ID，例如我的ID是marcovaldong，那么我们就可以通过访问地址https://www.zhihu.com/people/marcovaldong来访问我的主页。个人主页中包含了居住地、所在行业、性别、教育情况、获得的赞数、感谢数、关注了哪些人、被哪些人关注等信息。因此，我首先介绍如何通过爬虫来获取某一个知乎用户的一些信息。下面的函数getUserInfo(userID)实现了爬取一个知乎用户的个人信息，我们传递给该用户一个用户ID，该函数就会返回一个list，其中包含昵称、ID、居住地、所在行业、性别、所在公司、职位、毕业学校、专业、赞同数、感谢数、提问数、回答数、文章数、收藏数、公共编辑数量、关注的人数、被关注的人数、主页被多少个人浏览过等19个数据。

```
def getUserInfo(userID):
    user_url = 'https://www.zhihu.com/people/' + userID
    response = s.get(user_url, headers=header_info)
    # print response
    soup = BeautifulSoup(response.content, 'lxml')
    name = soup.find_all('span', {'class': 'name'})[1].string
    # print 'name: %s' % name
    ID = userID
    # print 'ID: %s' % ID
    location = soup.find('span', {'class': 'location item'})
    if location == None:
        location = 'None'
    else:
        location = location.string
    # print 'location: %s' % location
    business = soup.find('span', {'class': 'business item'})
    if business == None:
        business = 'None'
    else:
        business = business.string
    # print 'business: %s' % business
    gender = soup.find('input', {'checked': 'checked'})
    if gender == None:
        gender = 'None'
    else:
```

```
gender = gender.replace('男')
# print 'gender: %s' % gender
employment = soup.find('span', {'class': 'employment item'})
if employment == None:
    employment = 'None'
else:
    employment = employment.string
# print 'employment: %s' % employment
position = soup.find('span', {'class': 'position item'})
if position == None:
    position = 'None'
else:
    position = position.string
# print 'position: %s' % position
education = soup.find('span', {"class": "education item"})
if education == None:
    education = 'None'
else:
    education = education.string
# print 'education: %s' % education
major = soup.find('span', {'class': 'education-extra item'})
if major == None:
    major = 'None'
else:
    major = major.string
# print 'major: %s' % major

agree = int(soup.find('span', {'class': 'zm-profile-header-user-agree'}).strong.string)
# print 'agree: %d' % agree
thanks = int(soup.find('span', {'class': 'zm-profile-header-user-thanks'}).strong.string)
# print 'thanks: %d' % thanks
infolist = soup.find_all('a', {'class': 'item'})
asks = int(infolist[1].span.string)
# print 'asks: %d' % asks
answers = int(infolist[2].span.string)
# print 'answers: %d' % answers
posts = int(infolist[3].span.string)
# print 'posts: %d' % posts
collections = int(infolist[4].span.string)
# print 'collections: %d' % collections
logs = int(infolist[5].span.string)
# print 'logs: %d' % logs
followees = int(infolist[len(infolist)-2].strong.string)
# print 'followees: %d' % followees
followers = int(infolist[len(infolist)-1].strong.string)
# print 'followers: %d' % followers
scantime = int(soup.find_all('span', {'class': 'zg-gray-normal'}))
[len(soup.find_all('span', {'class': 'zg-gray-normal'}))-1].strong.string
# print 'scantime: %d' % scantime

info = (name, ID, location, business, gender, employment, position,
        education, major, agree, thanks, asks, answers, posts,
        collections, logs, followees, followers, scantime)
return info
```

```
if __name__ == '__main__':
    login()
    userID = 'marcovaldong'
    info = get.userInfo(userID)
    print 'The information of ' + userID + ' is: '
    for i in range(len(info)):
        print info[i]
```

下图是我的主页的部分截图，从上面可以看到这19个数据，下面第二张图是终端上显示的我的这19个数据，我们可以作个对照，看看是否全部抓取到了。这个函数我用了很长时间来调试，因为不同人的主页的信息完整程度是不同的，如果你在使用过程中发现了错误，欢迎告诉我。

The screenshot shows a user profile on Zhihu. The profile card includes:

- Profile picture and name: 马可瓦多
- GitHub link: marcovaldong.github.io/
- Location: 北京 - 现居地
- Education: 北京邮电大学 - 计算机科学与技术
- Recent activity: 最近详细资料, 不了解, 不要谈.
- Statistics: 29 人关注, 4 人关注者, 14 个专栏, 53 个话题.
- Personal bio: 刘春山。知乎指南。建议反馈。启动应用加入知乎。知乎热议。联系我们 0/2016 知乎

Below the profile card, there is a list of recent questions asked by the user:

- 游匣7559无法安装linux,怎么解决? (0 赞同)
- Python三维可视化，机器学习代码，包出问题？ (0 赞同)
- 如何看待北京顺丰快递小哥刮车被车主打死事件？ (0 赞同)

```
marcovaldo@ubuntu: ~/Lab/spider
b66\u4e0e\u6280\u672f', 12, 7, 3, 12, 0, 16, 12, 29, 4, 186)
marcovaldo@ubuntu:~/Lab/spider$ python zhihu.py
19
marcovaldo@ubuntu:~/Lab/spider$ python zhihu.py
马可瓦多
marcovald
北京
填写行业
male
填写公司信息
填写职位
北京邮电大学
计算机科学与技术
12
7
3
12
0
16
12
29
4
186
marcovaldo@ubuntu:~/Lab/spider$
```

## 获取某个答案的所有点赞者名单

知乎上有一个问题是如何写个爬虫程序扒下知乎某个回答所有点赞用户名单？，我参考了段小草的这个答案如何入门Python爬虫，然后有了下面的这个函数。

这里先来大概的分析一下整个流程。我们要知道，知乎上的每一个问题都有一个唯一ID，这个可以从地址中看出来，例如问题2015年有哪些书你读过以后觉得名不符实？的地址为 <https://www.zhihu.com/question/38808048>，其中38808048就是其ID。而每一个问题下的每一个答案也有一个唯一ID，例如该问题下的最高票答案2015年有哪些书你读过以后觉得名不符实？ - 余悦的回答 - 知乎的地址链接为 <https://www.zhihu.com/question/38808048/answer/81388411>，末尾的81388411就是该答案在该问题下的唯一ID。不过我们这里用到的不是这两个ID，而是我们在抓取点赞者名单时的唯一ID，此ID的获得方法是这样：例如我们打算

抓取如何评价《人间正道是沧桑》这部电视剧？ - 老编辑的回答 - 知乎的点赞者名单，首先打开firebug，点击“5321人赞同”时，firebug会抓取到一个“GET voters\_profile”的一个包，把光标放在上面，会看到一个链接 [https://www.zhihu.com/answer/5430533/voters\\_profile](https://www.zhihu.com/answer/5430533/voters_profile)，其中的5430533才是我们在抓取点赞者名单时用到的一个唯一ID。注意此ID只有在答案被赞过后才有。(在这安利一下《人间正道是沧桑》这部电视剧，该剧以杨立青三兄妹的恩怨情仇为线索，从大革命时期到解放战争，比较全面客观的展现了国共两党之间的主义之争，每一次看都会新的认识和体会。)

在拿到唯一ID后，我们用requests模块去get到知乎返回的信息，其中有一个json语句，该json语句中包含点赞者的信息。另外，我们在网页上浏览点赞者名单时，一次只能看到20条，每次下拉到名单底部时又加载出20条信息，再加载20条信息时所用的请求地址也包含在前面的json语句中。因此我们需要从json语句中提取出点赞着信息和下一个请求地址。在网页上浏览点赞者名单时，我们可以看到点赞者的昵称、头像、获得了多少赞同和感谢，以及提问和回答的问题数量，这里我提取了每个点赞者的昵称、主页地址（也就是用户ID）、赞同数、感谢数、提问数和回答数。关于头像的提取，我会在下面的函数中实现。

在提取到点赞者名单后，我将者信息保存了以唯一ID命名的txt文件中。下面是函数的具体实现。

```
Zhihu = "http://www.zhihu.com"
def get_voters(ans_id):
    # 直接输入问题id(这个id在点击“等人赞同”时可以通过监听网络得到)，关注者保存在以问题id命名的.txt文件中
    login()
    file_name = str(ans_id) + '.txt'
    f = open(file_name, 'w')
    source_url = Zhihu + '/answer/' + str(ans_id) + '/voters_profile'
    source = s.get(source_url, headers=header_info)
    print source
    content = source.content
    print content  # json语句
    data = json.loads(content)  # 包含总赞数，一组点赞者的信息，指向下一组点赞者的资源等的数据
    # 打印总赞数
    txt1 = '总赞数：'
    print txt1.decode('utf-8')
    total = data['paging'][0]['total']  # 总赞数
    print data['paging'][0]['total']  # 总赞数
    # 通过分析：每一组资源包含10个点赞者的信息（当然，最后一组可能少于10个），所以需要循环遍历
    nextsource_url = source_url  # 从第0组点赞者开始剖析
    num = 0
```

```

while nextsource_url!=Zhihu:
    try:
        nextsource = s.get(nextsource_url, headers=header_info)
    except:
        time.sleep(2)
        nextsource = s.get(nextsource_url, headers=header_info)
    # 解析出点赞者的信息
    nextcontent = nextsource.content
    nextdata = json.loads(nextcontent)
    # 打印每个点赞者的信息
    # txt2 = '打印每个点赞者的信息'
    # print txt2.decode('utf-8')
    # 提取每个点赞者的基本信息
    for each in nextdata['payload']:
        num += 1
        print num
        try:
            soup = BeautifulSoup(each, 'lxml')
            tag = soup.a
            title = tag['title']      # 点赞者的用户名
            href = 'http://www.zhihu.com' + str(tag['href'])      # 点赞者的地址
            # 获取点赞者的数据
            list = soup.find_all('li')
            votes = list[0].string  # 点赞者获得的赞同
            tks = list[1].string  # 点赞者获得的感谢
            ques = list[2].string  # 点赞者提出的问题数量
            ans = list[3].string  # 点赞者回答的问题数量
            # 打印点赞者信息
            string = title + ' ' + href + ' ' + votes + tks + ques + ans
            f.write(string + '\n')
            print string
        except:
            txt3 = '有点赞者的信息缺失'
            f.write(txt3.decode('utf-8') + '\n')
            print txt3.decode('utf-8')
            continue
    # 解析出指向下一组点赞者的资源
    nextsource_url = Zhihu + nextdata['paging']['next']
f.close()

```

注意，点赞者名单中会有匿名用户，或者有用户被注销，这时我们抓取不到此用户的信息，我这里在txt文件中添加了一句“有点赞者的信息缺失”。

使用同样的方法，我们就可以抓取到一个用户的关注者名单和被关注者名单，下面列出了这两个函数。但是关注者名单抓取函数有一个问题，每次使用其抓取大V的关注者名单时，当抓取到第10020个follower的时候程序就会报错，好像知乎有访问限制一般。这个问题，我还没有找到解决办法，希望有solution的告知一下。因为没有看到有用户关注10020+个人，

```
def get_followees(username):
```

```

# 直接输入用户名，关注者保存在以用户名命名的.txt文件中
followers_url = 'http://www.zhihu.com/people/' + username + '/followees'
file_name = username + '.txt'
f = open(file_name, 'w')
data = s.get(followers_url, headers=header_info)
print data # 访问服务器成功，返回<response 200>
content = data.content # 提取出html信息
soup = BeautifulSoup(content, "lxml") # 对html信息进行解析
# 获取关注者数量
totalsen = soup.select('span[class="zm-profile-section-name"]')
total = int(str(totalsen[0]).split(' ')[4]) # 总的关注者数量
txt1 = '总的的关注者人数：'
print txt1.decode('utf-8')
print total
follist = soup.select('div[class="zm-profile-card"]') # 记录有关关注者信息的list
num = 0 # 用来在下面显示正在查询第多少个关注者
for follower in follist:
    tag = follower.a
    title = tag['title'] # 用户名
    href = 'http://www.zhihu.com' + str(tag['href']) # 用户地址
    # 获取用户数据
    num += 1
    print 'No %d % (num, num / float(total))'
    # Alist = follower.findall(has_attrs)
    Alist = follower.find_all('a', {'target': '_blank'})
    votes = Alist[0].string # 点赞者获取的赞同
    tks = Alist[1].string # 点赞者获取的感谢
    ques = Alist[2].string # 点赞者提出的问题数量
    ans = Alist[3].string # 点赞者回答的问题数量
    # 打印关注者信息
    string = title + ' ' + href + ' ' + votes + tks + ques + ans
    try:
        print string.decode('utf-8')
    except:
        print string.encode('gbk', 'ignore')
    f.write(string + '\n')

# 遍历次数
n = total/20-1 if total/20.0-total/20 == 0 else total/20
for i in range(1, n+1, 1):
    # if num%30 == 0:
    #     time.sleep(1)
    # if num%50 == 0:
    #     time.sleep(2)
    raw_hash_id = re.findall('hash_id(.*)', content)
    hash_id = raw_hash_id[0][14:46]
    _xsrft = xsrf
    offset = 20*i
    params = json.dumps({'offset': offset, 'order_by': 'created', 'hash_id': hash_id})
    payload = {"method": "next", "params": params, "_xsrft": _xsrft}
    click_url = 'http://www.zhihu.com/node/ProfileFolloweesListV2'
    data = s.post(click_url, data=payload, headers=header_info)
    # print data

```

```
source = json.loads(data.content)
for follower in source['msg']:
    soup1 = BeautifulSoup(follower, 'lxml')
    tag =soup1.a
    title = tag['title']      # 用户名
    href = 'http://www.zhihu.com' + str(tag['href'])      # 用户地址
    # 获取用户数据
    num +=1
    print '%d  %f' % (num, num/float(total))
    # Alist = soup1.find_all(has_attrs)
    Alist = soup1.find_all('a', {'target': '_blank'})
    votes = Alist[0].string # 点赞者获取的赞同
    tks = Alist[1].string # 点赞者获得的感谢
    ques = Alist[2].string # 点赞者提出的问题数量
    ans = Alist[3].string # 点赞者回答的问题数量
    # 打印关注者信息
    string = title + ' ' + href + ' ' + votes + tks + ques + ans
    try:
        print string.decode('utf-8')
    except:
        print string.encode('gbk','ignore')
    f.write(string + '\n')
f.close()
```

因此抓取被关注者名单函数暂时未发现报错。

## 提取用户头像

---

再往下就是抓取用户头像了，给出某个唯一ID，下面的函数自动解析其主页，从中解析出该用户头像地址，抓取到图片并

```
def get_avatar(userId):
    url = 'https://www.zhihu.com/people/' + userId
    response = s.get(url, headers=header_info)
    response = response.content
    soup = BeautifulSoup(response, 'lxml')
    name = soup.find_all('span', {'class': 'name'})[1].string
    # print name
    temp = soup.find('img', {'alt': name})
    avatar_url = temp['src'][0:-6] + temp['src'][-4:]
    filename = 'pics/' + userId + temp['src'][-4:]
    f = open(filename, 'wb')
    f.write(requests.get(avatar_url).content)
    f.close()
```

保存到本地文件，文件以用户唯一ID命名。

结合其他函数，我们就可以抓取到某个答案下所有点赞者的头像，某个大V所有followers的头像等。

## 抓取某个问题的所有答案

给出某个唯一ID，下面的函数帮助爬取到该问题下的所有答案。注意，答案内容只抓取文字部分，图片省略，答案保存在txt文件中，txt文件以答主ID命名。

```
def get_answer(questionID):
    url = 'http://www.zhihu.com/question/' + str(questionID)
    data = s.get(url, headers=header_info)
    soup = BeautifulSoup(data.content, 'lxml')
    # print str(soup).encode('gbk', 'ignore')
    title = soup.title.string.split('\n')[2]    # 问题题目
    path = title
    if not os.path.isdir(path):
        os.mkdir(path)
    description = soup.find('div', {'class': 'zm-editable-content'}).strings    # 问题描述，可能多行
    file_name = path + '/description.txt'
    fw = open(file_name, 'w')
```

```
for each in description:
    each = each + '\n'
    fw.write(each)
# description = soup.find('div', {'class': 'zm-editable-content'}).get_text() # 问题描述
# s.string属性返回None(可能是因为有换行符在内的缘故),调用get_text()方法得到了文本,但换行丢了
answer_num = int(soup.find('h3', {'id': 'zh-question-answer-num'}).string.split(' ')[0]) # 答案数量
num = 1
index = soup.find_all('div', {'tabindex': '-1'})
for i in range(len(index)):
    print ('Scraping the ' + str(num) + 'th answer.....').encode('gbk', 'ignore')
    # print ('正在抓取第' + str(num) + '个答案.....').encode('gbk', 'ignore')
    try:
        a = index[i].find('a', {'class': 'author-link'})
        title = str(num) + '__' + a.string
        href = 'http://www.zhihu.com' + a['href']
    except:
        title = str(num) + '__匿名用户'
    answer_file_name = path + '/' + title + '.txt'
    fr = open(answer_file_name, 'w')
    try:
        answer_content = index[i].find('div', {'class': 'zm-editable-content clearfix'}).strings
    except:
        answer_content = ['作者修改内容通过后,回答会重新显示。如果一周内未得到有效修改,回答会自动折叠。']
    for content in answer_content:
        fr.write(content + '\n')
    num += 1

_xsrf = xsrf
url_token = re.findall('url_token(.*)', data.content)[0][8:16]
# 循环次数
n = answer_num/10-1 if answer_num/10.0-answer_num/10 == 0 else answer_num/10
for i in range(1, n+1, 1):
    # _xsrft = xsrf
    # url_token = re.findall('url_token(.*)', data.content)[0][8:16]
    offset = 10*i
    params = json.dumps({'url_token': url_token, 'pagesize': 10, 'offset': offset})
    payload = {"method": "next", "params": params, "_xsrft": _xsrft}
    click_url = 'https://www.zhihu.com/node/QuestionAnswerListV2'
    data = s.post(click_url, data=payload, headers=header_info)
    data = json.loads(data.content)
    for answer in data['msg']:
        print ('Scraping the ' + str(num) + 'th answer.....').encode('gbk', 'ignore')
        # print ('正在抓取第' + str(num) + '个答案.....').encode('gbk', 'ignore')
        soupl = BeautifulSoup(answer, 'lxml')
        try:
            a = soupl.find('a', {'class': 'author-link'})
            title = str(num) + '__' + a.string
            href = 'http://www.zhihu.com' + a['href']
        except:
            title = str(num) + '__匿名用户'
        answer_file_name = path + '/' + title + '.txt'
        fr = open(answer_file_name, 'w')
        frv:
```

```
    answer_content = soup1.find('div', {'class': 'zm-editable-content clearfix'}).strings
except:
    answer_content = ['作者修改内容通过后，回答会重新显示。如果一周内未得到有效修改，回答会自动折叠。']
for content in answer_content:
    fr.write(content + '\n')
num += 1
```

## 数据库存取数据

在完成了上面的这些功能后，下一步要做的是将用户信息保存在数据库中，方便数据的读取使用。我刚刚接触了一下sqlite3，仅仅实现了将用户信息存储在表格中。

```
def get_followeesInfo_toDB(userID):
    # 准备好sqlite3数据库，当插入到数据库时，加入表格中
    conn = sqlite3.connect("Zhihu.db")
    curs = conn.cursor()
    curs.execute("create table if not exists userinfo(name TEXT, ID TEXT PRIMARY KEY, location TEXT, business TEXT, "
                "gender TEXT, employment TEXT, position TEXT, education TEXT, major TEXT, "
                "agree INTEGER, thanks INTEGER, asks INTEGER, answers INTEGER, posts INTEGER, "
                "collections INTEGER, logs INTEGER, followees INTEGER, followers INTEGER, "
                "scantime INTEGER)")
    followees_url = 'http://www.zhihu.com/people/' + userID + '/followees'
    file_name = userID + '.txt'
    f = open(file_name, 'w')
    data = s.get(followees_url, headers=header_info)
    print data # 访问服务器成功，返回<response 200>
    content = data.content # 撕取出html信息
    soup = BeautifulSoup(content, "lxml") # 对html信息进行解析
    # 获取关注者数量
    totalsen = soup.select('span[class="zm-profile-section-name"]')
    total = int(str(totalsen[0]).split(' ')[4]) # 总的关注者数量
    txt1 = '总的关注者人数：'
    print txt1.decode('utf-8')
    print total
    follist = soup.select('div[class="zm-profile-card"]') # 记录有关注者信息的list
    num = 0 # 用来在下面显示正在遍历第多少个关注者
    for follower in follist:
        tag = follower.a
        title = tag['title'] # 用户名
        href = 'http://www.zhihu.com' + str(tag['href']) # 用户地址
        # 获取用户数据
        num += 1
        print '%d %f' % (num, num / float(total))
        # Alist = follower.find_all(attrs)
        Alist = follower.find_all('a', {'target': '_blank'})
        votes = Alist[0].string # 点赞数获取的赞同
        tks = Alist[1].string # 占点赞数的感谢
```

```
ques = Alist[2].string # 点赞者提出的问题数量
ans = Alist[3].string # 点赞者回答的问题数量
# 打印关注者信息
string = title + ' ' + href + ' ' + votes + tks + ques + ans
try:
    print string.decode('utf-8')
except:
    print string.encode('gbk', 'ignore')
f.write(string + '\n')
if title != '[已重置]':
    # 获取该followee的基本信息，存入数据库表格
    print 'Analysising the data of this user...'
    ID = href[28:]
    try:
        curs.execute("insert or ignore into userinfo values (?, ?, ?, ?, ?, ?, ?, ?,
                    ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)", get.userInfo(ID))
    except:
        print "This user account's state is abnormal..."
    else:
        print 'This user account has been disabled...'
    # print get.userInfo(ID)

# 循环次数
n = total / 20 - 1 if total / 20.0 - total / 20 == 0 else total / 20
for i in range(1, n + 1, 1):
    # if num%30 == 0:
    #     time.sleep(1)
    # if num%50 == 0:
    #     time.sleep(3)
    raw_hash_id = re.findall('hash_id(.*)', content)
    hash_id = raw_hash_id[0][14:46]
    _xsrft = xsrf
    offset = 20 * i
    params = json.dumps({'offset': offset, 'order_by': 'created', 'hash_id': hash_id})
    payload = {'method': 'next', 'params': params, '_xsrft': _xsrft}
    click_url = 'http://www.zhihu.com/node/ProfileFolloweesListV2'
    data = s.post(click_url, data=payload, headers=header_info)
    # print data
    source = json.loads(data.content)
    for follower in source['msg']:
        soup1 = BeautifulSoup(follower, 'lxml')
        tag = soup1.a
        title = tag['title'] # 用户名
        href = 'http://www.zhihu.com' + str(tag['href']) # 用户地址
        # 获取用户数据
        num += 1
        print '%d %f' % (num, num / float(total))
        # Alist = soup1.findall(has_attrs)
        Alist = soup1.findAll('a', {'target': '_blank'})
        votes = Alist[0].string # 点赞者获得的赞同
        tks = Alist[1].string # 点赞者获取的感谢
        ques = Alist[2].string # 点赞者提出的问题数量
```



《Linux云计算及运维架构师高薪实战班》2018年11月26日即将开课中，**120天冲击Linux运维年薪30万**，改变速约~~~~~



马哥实战学院

入门课程免费学习

小程序

\*声明：推送内容及图片来源于网络，部分内容会有所改动，版权归原作者所有，如来源信息有误或侵犯权益，请联系我们删除或授权事宜。

- END -



# Python福利包

主讲人：上市公司十年开发经理

福利1：15册Python入门书籍

福利2：30集Python入门视频

福利3：50个Python商业项目源代码



长按识别二维码，即刻获取



[Read more](#)