

第 7 关、爬取知乎文章

1、项目实操

1-1、确定目标

(1) 目标网站：<https://www.zhihu.com/people/zhang-jia-wei/posts?page=1>（按点赞数给文章排序）；

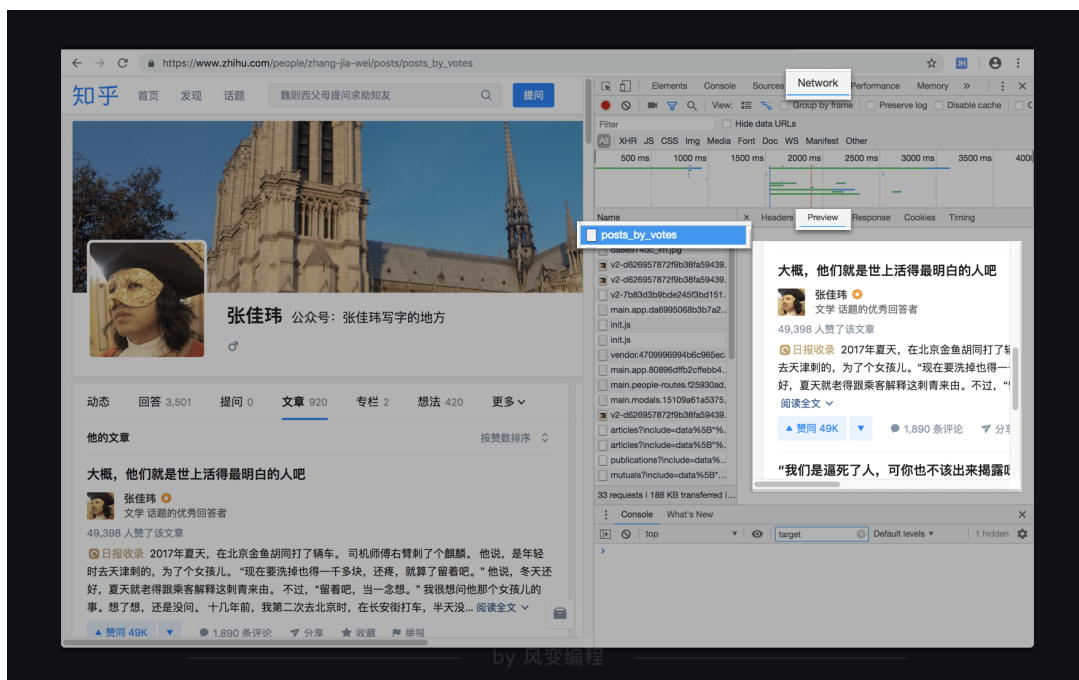


(2) 项目目标：爬取知乎大v张佳玮的文章“标题”、“摘要”、“链接”，并存储到本地文件。

1-2、过程分析

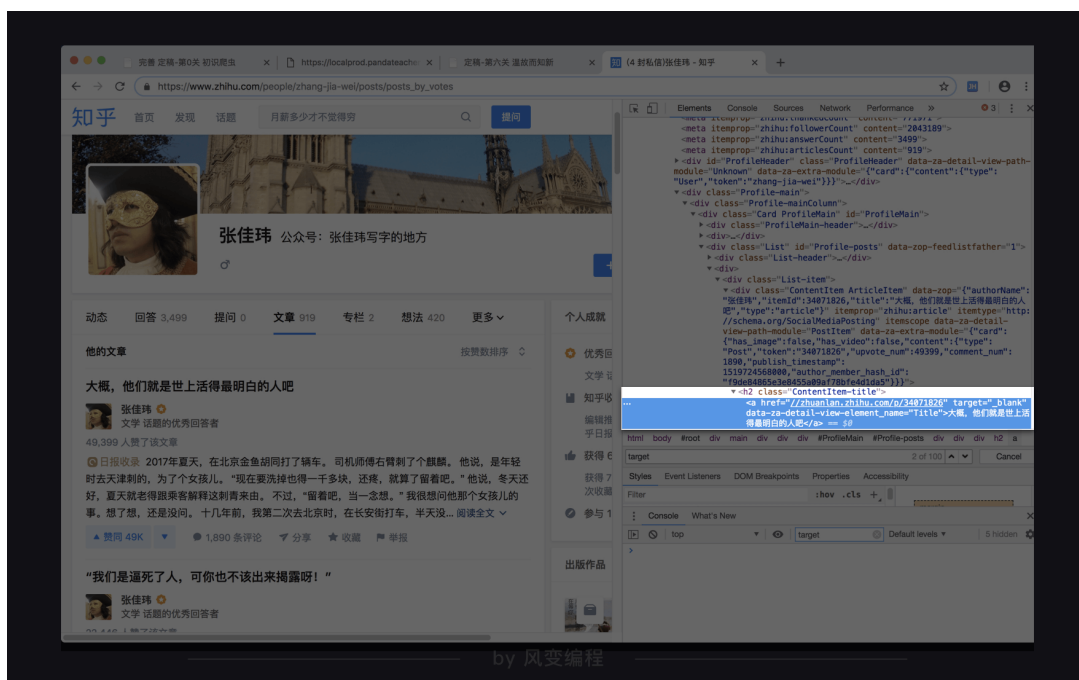
(1) 确定数据所在页面

①、右键选择【检查】；点击【Network】，选择【All】（而非 XHR）；③、刷新网页，点进去第 0 个请求: posts_by_votes；④、点击【Preview】。



(2) 确定数据所在位置

文章标题对应的就是 `<a>` 元素，这个 `<a>` 标签的上一个层级是 `<h2>` 标签，并且有 `class="ContentItem-title"`，这个属性可以帮我们精准定位目标数据。



(3) 数据获取思路

获取数据——用 `requests` 库；解析数据——用 `BeautifulSoup` 库；提取数据——用 `BeautifulSoup` 里的 `find_all()`，翻页的话观察第一页，到最后一页的网址特征，写循环；存储数据——用 `csv` 和 `openpyxl` 都可以。

1-3、代码实现

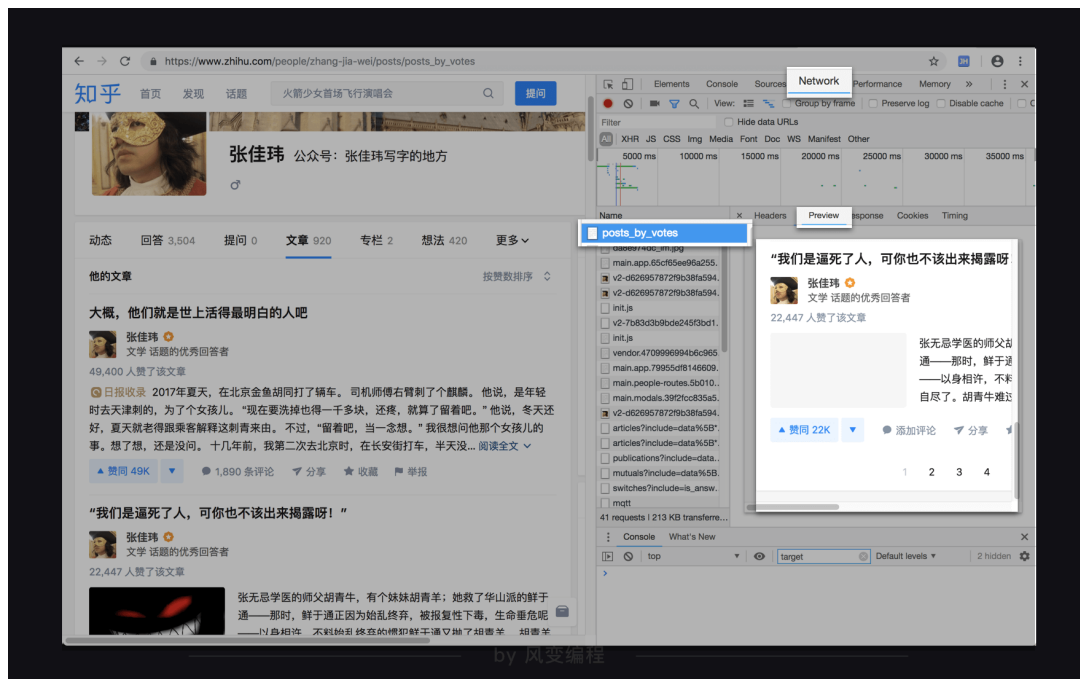
```
1 import requests
2 from bs4 import BeautifulSoup
3 #引入request和bs
```

```

4 url='https://www.zhihu.com/people/zhang-jia-wei/posts/posts_by_votes?
  page=1'
5 headers={'user-agent':'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'}
6 #使用headers是一种默认的习惯，默认你已经掌握啦~
7 res=requests.get(url,headers=headers)
8 #发起请求，将响应的结果赋值给变量res。
9 print(res.status_code)
10 #检查状态码
11 bstitle=BeautifulSoup(res.text,'html.parser')
12 #用bs进行解析
13 title=bstitle.findAll(class_='ContentItem-title')
14 #提取我们想要的标签和里面的内容
15 print(title)
16 #打印title

```

结果只有两个标题，再观察我们的 posts_by_votes 文件，发现 HTML 里面只放了两篇文章的数据。



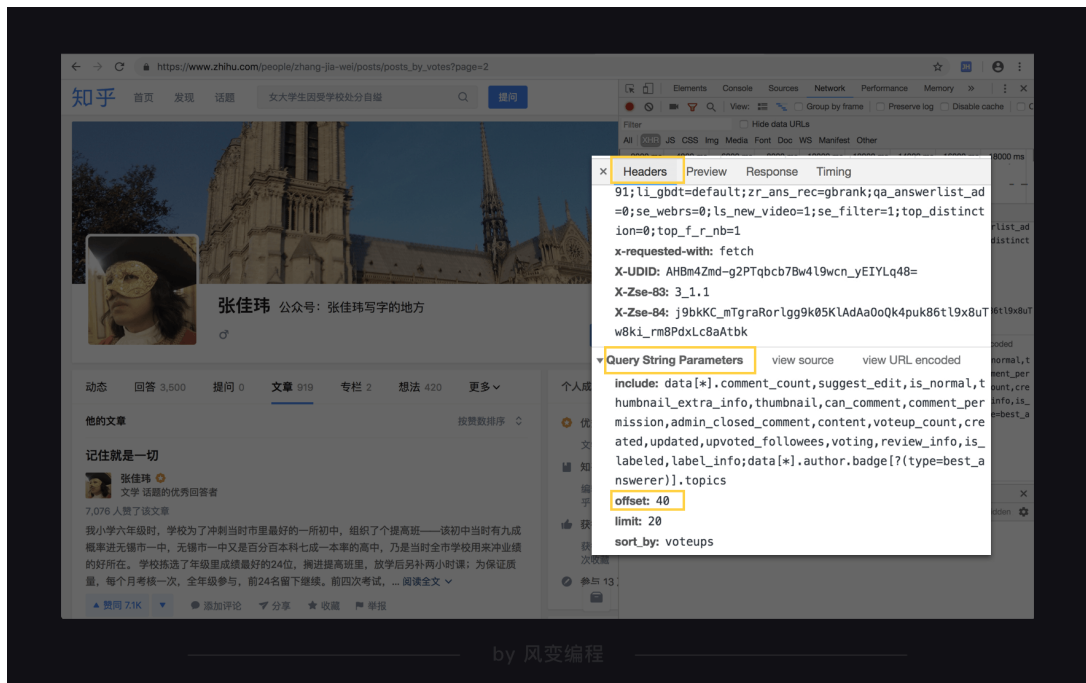
1-4、重新分析过程

(1) 确定数据所在页面

打开 Network，点开 XHR，同时刷新页面，看到出现了很多个请求，主要查看 articles 的请求。可以发现文章的链接、文章的摘要也是一样放在 articles 的请求里。

(2) 数据所在链接的请求参数

观察第 1 页对第 2 页的请求，和第 2 页里对第 3 页请求的参数区别，是在 headers 里面的 query string parameters 里面。



发现除了offset都一样，offset代表起始值，limit表示加载的限制数，通过循环我们可以爬到所有页数的内容了。

【爬取思路】

获取数据 去拿请求里面的headers里面的request url，操作为response对象

解析数据 使用response.json()去解析这个json数据

提取数据 根据列表和字典的相关知识，可以提取出我们要的文章标题、摘要和链接

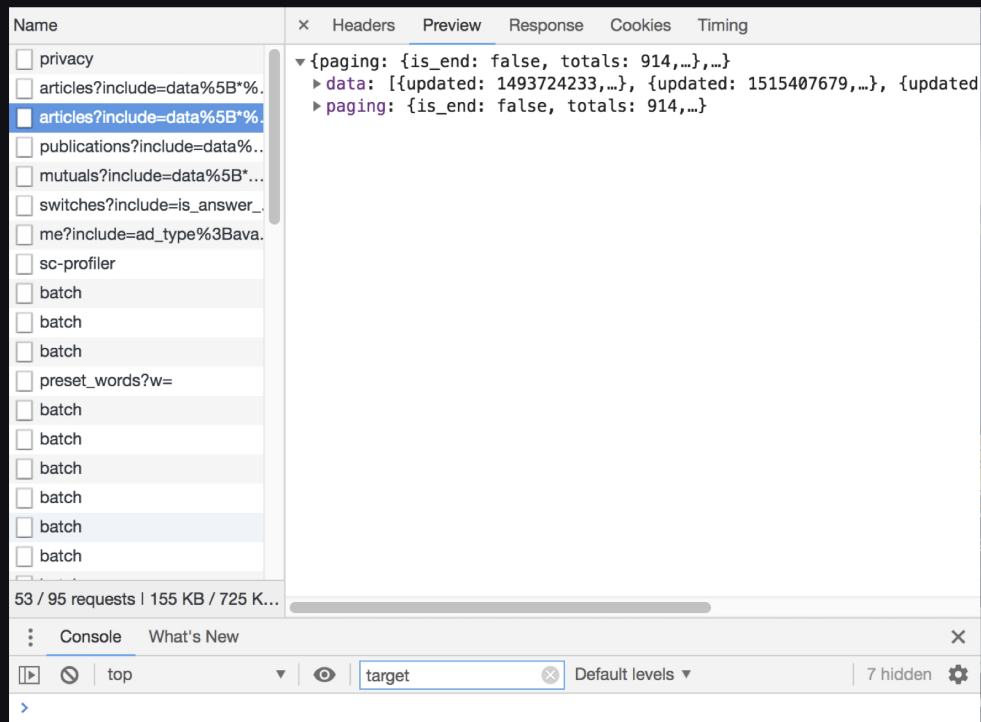
拿完一页后，对比第一页和第二页的request url，发现区别，然后用循环写入

存储数据 使用csv或者openpyxl把数据存储下来

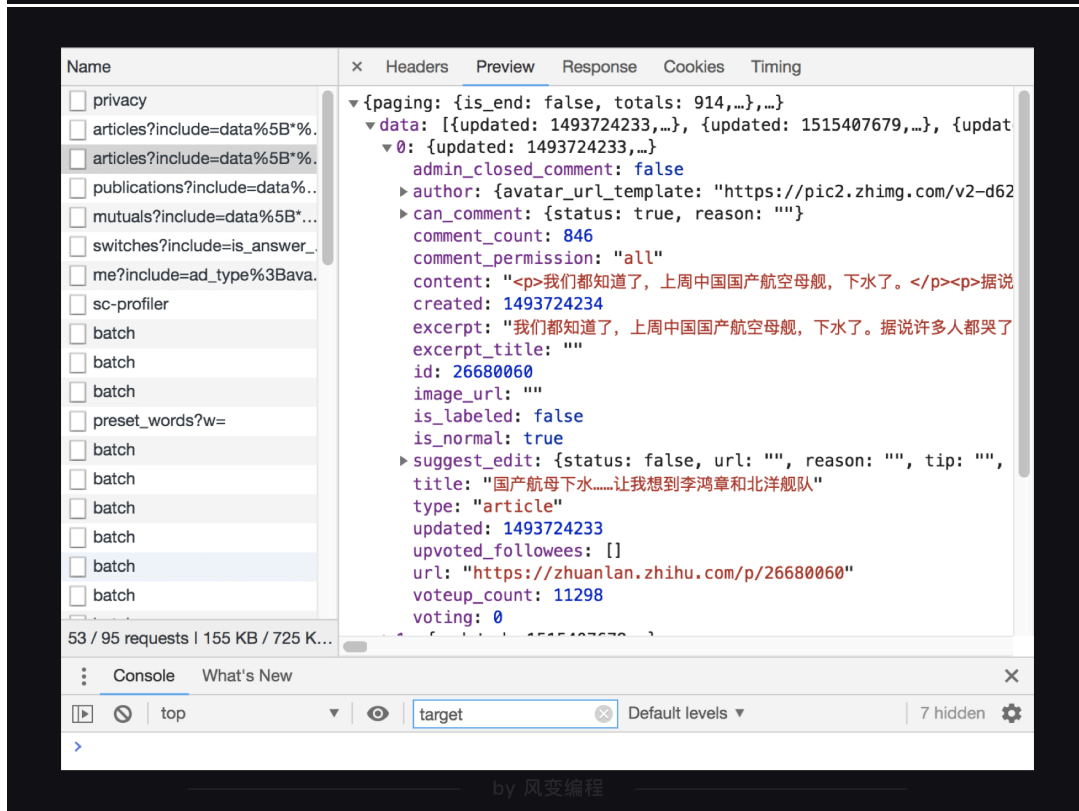
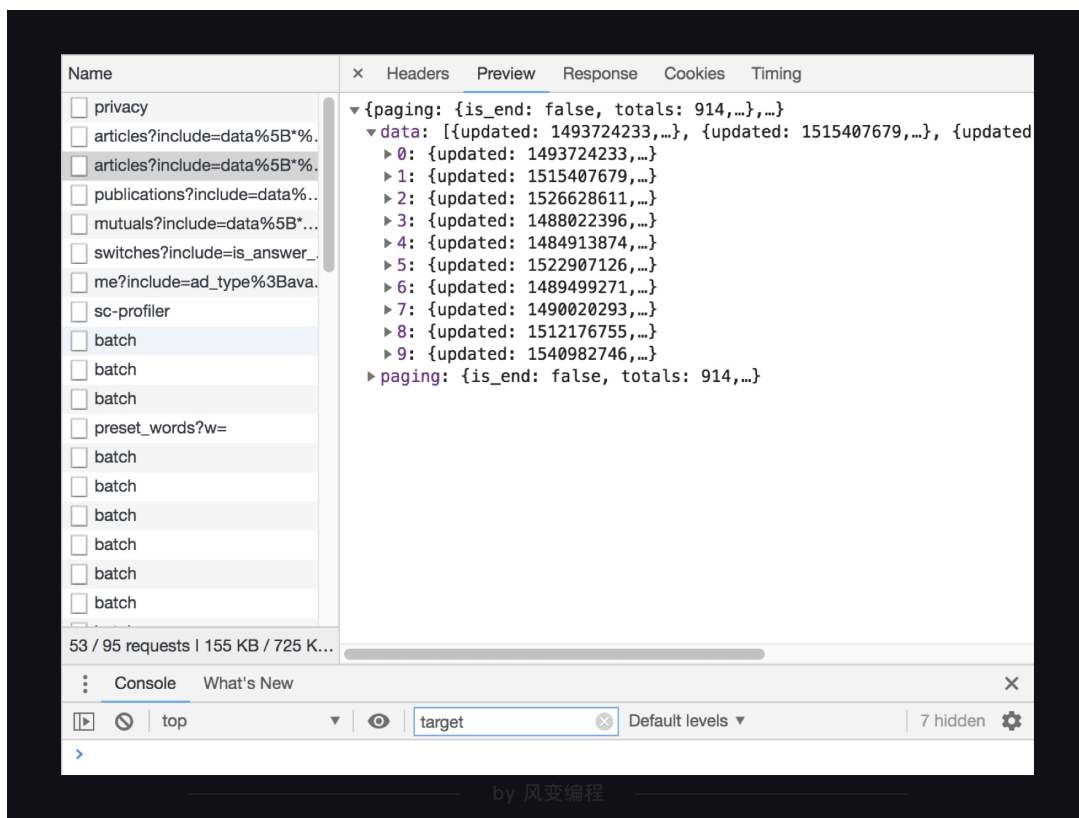
by 风变编程

(3) 确定数据所在位置

在preview里面一层层看看这个json文件，到底是怎么一个结构。下面三张图依次是从外到内的数据展示：



by 风变编程



最外层是一个很大的字典，里面有两大元素，data:和paging:，这两大元素又是键值对应的字典形式，data这个键所对应的值是一个列表，里面有10元素，每个元素又是字典形式。



1-5、重新实现代码

1-5-1、获取第一页的数据

```
1 import requests
2 #引入requests
3 headers={'user-agent':'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6)
4   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'}
5 #封装headers
6 url='https://www.zhihu.com/api/v4/members/zhang-jia-wei/articles?'
7 #写入网址
8 params={
9     'include':'data[*].comment_count,suggest_edit,is_normal,thumbnail_extra_in
10    fo,thumbnail,can_comment,comment_permission,admin_closed_comment,content,v
11    oteup_count,created,updated,upvoted_followees,voting,review_info,is_labele
12    d,label_info;data[*].author.badge[?(type=best_answerer)].topics',
13     'offset':'10',
14     'limit':'20',
15     'sort_by':'voteups',
16 }
17 #封装参数
18 res=requests.get(url,headers=headers,params=params)
19 #发送请求，并把响应内容赋值到变量res里面
20 print(res.status_code)
21 #确认这个Response对象状态正确
22 articles=res.json()
23 #用response.json()方法去解析数据，并赋值到变量articles上面，此时的articles是一个
24 print(articles)
25 #打印这个json文件
26 data=articles['data']
27 #取出键为data的值。
28 for i in data:
29     print(i['title'])
30     print(i['url'])
```



```
27     print(i['excerpt'])
28     #遍历列表，拿到的是列表里的每一个元素，这些元素都是字典，再通过键把值取出来
```

1-5-2、获取所有页面数据

```
1  import requests
2  headers={'user-agent':'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6)
   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'}
3  url='https://www.zhihu.com/api/v4/members/zhang-jia-wei/articles?'
4  articlelist=[]
5  #建立一个空列表，以待写入数据
6  offset=0
7  #设置offset的起始值为0
8  while True:
9      params={
10
11          'include':'data[*].comment_count,suggest_edit,is_normal,thumbnail_extra_in
   fo,thumbnail,can_comment,comment_permission,admin_closed_comment,content,v
   oteup_count,created,updated,upvoted_followees,voting,review_info,is_labele
   d,label_info;data[*].author.badge[?(type=best_answerer)].topics',
12          'offset':str(offset),
13          'limit':'20',
14          'sort_by':'voteups',
15      }
16      #封装参数
17      res=requests.get(url,headers=headers,params=params)
18      #发送请求，并把响应内容赋值到变量res里面
19      articles=res.json()
20      # print(articles)
21      data=articles['data']
22      #定位数据
23      for i in data:
24          list1=[i['title'],i['url'],i['excerpt']]
25          #把数据封装成列表
26          articlelist.append(list1)
27          offset=offset+20
28          #在while循环内部，offset的值每次增加20
29          if offset>40:
30              break
31          #如果offset大于40，即爬了两页，就停止
32          #if articles['paging']['is_end'] == True:
33          #如果键is_end所对应的值是True，就结束while循环。
34          #break
35      print(articlelist)
36      #打印看看
```

1-5-3、存储数据

```
1  import requests
2  import csv
```



```

3 #引用csv。
4 csv_file=open('articles.csv','w',newline='',encoding='utf-8')
5 #调用open()函数打开csv文件，传入参数：文件名“articles.csv”、写入模式“w”、
  newline=''。
6 writer = csv.writer(csv_file)
7 # 用csv.writer()函数创建一个writer对象。
8 list2=['标题','链接','摘要']
9 #创建一个列表
10 writer.writerow(list2)
11 #调用writer对象的writerow()方法，可以在csv文件里写入一行文字 “标题”和“链接”和“摘要”。
12
13 headers={'user-agent':'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'}
14 url='https://www.zhihu.com/api/v4/members/zhang-jia-wei/articles?'
15 offset=0
16 #设置offset的起始值为0
17 while True:
18     params={
19
20         'include':'data[*].comment_count,suggest_edit,is_normal,thumbnail_extra_in
  fo,thumbnail,can_comment,comment_permission,admin_closed_comment,content,v
  oteup_count,created,updated,upvoted_followees,voting,review_info,is_labele
  d,label_info;data[*].author.badge[?(type=best_answerer)].topics',
21         'offset':str(offset),
22         'limit':'20',
23         'sort_by':'voteups',
24     }
25     #封装参数
26     res=requests.get(url,headers=headers,params=params)
27     #发送请求，并把响应内容赋值到变量res里面
28     articles=res.json()
29     print(articles)
30     data=articles['data']
31     #定位数据
32     for i in data:
33         list1=[i['title'],i['url'],i['excerpt']]
34         #把目标数据封装成一个列表
35         writer.writerow(list1)
36         #调用writerow()方法，把列表list1的内容写入
37         offset=offset+20
38         #在while循环内部，offset的值每次增加20
39         if offset > 40:
40             break
41     csv_file.close()
42     #写入完成后，关闭文件就大功告成
43     print('okay')

```