

第 12 关、协程实践

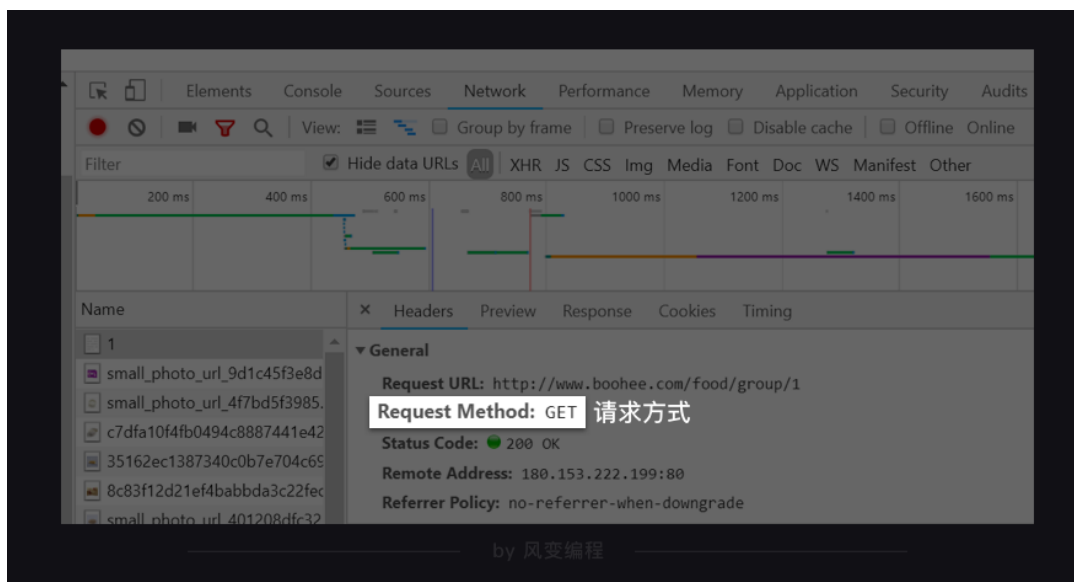
1、项目：薄荷网爬取食物热量

1-1、明确目标

- (1) 目标网站：<http://www.boohee.com/food/>;
- (2) 网站协议：<http://www.boohee.com/food/robots.txt>（目标网站 + robots.txt 可查看目标网站的页面爬取许可）；
- (3) 项目目标：用多协程爬取薄荷网的食物热量信息（包含食物名、热量、食物详情页面链接）。

1-2、过程分析

- (1) 确定数据所在页面
 - 右击打开“检查”工具，并点击 Network，然后刷新页面。点击第 0 个请求 1，看 Response 能在 Response 里找到食物的信息，说明我们想要的数据存在 HTML 里；
 - 再看请求 1 的 Headers，可以发现薄荷网的网页请求方式是 get（即通过 requests.get() 获取数据）。



- (2) 确定数据所在位置

①、每个常见食物分类的网址

- 网址的group参数代表着常见食物分类，后面的数字代表着这是第几个类；

常见食物分类的网址

类别	网址
第1类【谷薯芋、杂豆、主食】	http://www.boohee.com/food/group/1
第2类【蛋类、肉类及制品】	http://www.boohee.com/food/group/2
第3类【奶类及制品】	http://www.boohee.com/food/group/3
第4类【蔬果和菌藻】	http://www.boohee.com/food/group/4
第5类【坚果、大豆及制品】	http://www.boohee.com/food/group/5
第6类【饮料】	http://www.boohee.com/food/group/6
第7类【食用油、油脂及制品】	http://www.boohee.com/food/group/7
第8类【调味品】	http://www.boohee.com/food/group/8
第9类【零食、点心、冷饮】	http://www.boohee.com/food/group/9
第10类【其他】	http://www.boohee.com/food/group/10
第11类【菜肴】	http://www.boohee.com/food/view_menu

by 风变编程

- 除常见食物分类【菜肴】的网址与其他不同，前10个常见食物分类的网址都是：<http://www.boohee.com/food/group/+数字>
- ②、每一页食物的网址
- 只要改变网址 <http://www.boohee.com/food/group/1?page=> 中 page 后面的数字，就能实现翻页

【谷薯芋、杂豆、主食】的第2页食物记录的网址：

<http://www.boohee.com/food/group/1?page=2>

【谷薯芋、杂豆、主食】的第3页食物记录的网址：

<http://www.boohee.com/food/group/1?page=3>

【谷薯芋、杂豆、主食】的第4页食物记录的网址：

<http://www.boohee.com/food/group/1?page=4>

by 风变编程

(3) 数据获取思路

用 `find_all/find` 就能提取出 `<li class="item clearfix">...` 标签下的食物详情链接、名称和热量；



1-3、代码实现

1-3-1、导入库和模块

```
1 #导入所需的库和模块:
2
3 from gevent import monkey
4 monkey.patch_all()
5 #让程序变成异步模式。
6 import gevent,requests, bs4, csv
7 from gevent.queue import Queue
```

1-3-2、创建队列存储食物信息

```
1 work = Queue()
2 #创建队列对象，并赋值给work。
3
4 #前3个常见食物分类的前3页的食物记录的网址:
5 url_1 = 'http://www.boohie.com/food/group/{type}?page={page}'
6 for x in range(1, 4):
7     for y in range(1, 4):
```

```

8         real_url = url_1.format(type=x, page=y)
9         work.put_nowait(real_url)
10 #通过两个for循环，能设置分类的数字和页数的数字。
11 #然后，把构造好的网址用put_nowait方法添加进队列里。
12
13 #第11个常见食物分类的前3页的食物记录的网址：
14 url_2 = 'http://www.boohee.com/food/view_menu?page={page}'
15 for x in range(1,4):
16     real_url = url_2.format(page=x)
17     work.put_nowait(real_url)
18 #通过for循环，能设置第11个常见食物分类的食物的页数。
19 #然后，把构造好的网址用put_nowait方法添加进队列里。
20
21 print(work)
22 #打印队列

```

- 第 7 行用 Queue() 创建了空的队列；
- 第 12 – 23 行通过两个 for 循环，构造了前 3 个常见食物分类的前 3 页的食物记录的网址（其中由于第 11 个常见食物分类的网址比较特殊，要分开构造。然后把构造好的网址用 put_nowait 方法，都放进队列里。）；

1-3-3、多协程爬取数据

```

1 def crawler():
2     #定义crawler函数
3     headers = {
4         'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36'
5     }
6     #添加请求头
7     while not work.empty():
8         #当队列不是空的时候，就执行下面的程序。
9         url = work.get_nowait()
10        #用get_nowait()方法从队列里把刚刚放入的网址提取出来。
11        res = requests.get(url, headers=headers)
12        #用requests.get获取网页源代码。
13        bs_res = bs4.BeautifulSoup(res.text, 'html.parser')
14        #用BeautifulSoup解析网页源代码。
15        foods = bs_res.find_all('li', class_='item clearfix')
16        #用find_all提取出<li class="item clearfix">标签的内容。
17        for food in foods:
18            #遍历foods
19            food_name = food.find_all('a')[1]['title']
20            #用find_all在<li class="item clearfix">标签下，提取出第2个<a>元素
title属性的值，也就是食物名称。
21            food_url = 'http://www.boohee.com' + food.find_all('a')[1]
['href']
22            #用find_all在<li class="item clearfix">标签下，提取出第2个<a>元素
href属性的值，跟'http://www.boohee.com'组合在一起，就是食物详情页的链接。
23            food_calorie = food.find('p').text

```

```

24         #用find在<li class="item clearfix">标签下，提取<p>元素，再用text
方法留下纯文本，就提取出了食物的热量。
25         print(food_name)
26         #打印食物的名称。
27
28 tasks_list = []
29 #创建空的列表
30 for x in range(5):
31     #相当于创建了5个爬虫
32     task = gevent.spawn(crawler)
33     #用gevent.spawn()函数创建执行crawler()函数的任务。
34     tasks_list.append(task)
35     #往任务列表添加任务。
36 gevent.joinall(tasks_list)

```

1-3-4、csv 存储数据

```

1  from gevent import monkey
2  monkey.patch_all()
3  import gevent,requests, bs4, csv
4  from gevent.queue import Queue
5
6  work = Queue()
7  url_1 = 'http://www.boohee.com/food/group/{type}?page={page}'
8  for x in range(1, 4):
9      for y in range(1, 4):
10         real_url = url_1.format(type=x, page=y)
11         work.put_nowait(real_url)
12
13 url_2 = 'http://www.boohee.com/food/view_menu?page={page}'
14 for x in range(1,4):
15     real_url = url_2.format(page=x)
16     work.put_nowait(real_url)
17
18 def crawler():
19     headers = {
20         'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36'
21     }
22     while not work.empty():
23         url = work.get_nowait()
24         res = requests.get(url, headers=headers)
25         bs_res = bs4.BeautifulSoup(res.text, 'html.parser')
26         foods = bs_res.find_all('li', class_='item clearfix')
27         for food in foods:
28             food_name = food.find_all('a')[1]['title']
29             food_url = 'http://www.boohee.com' + food.find_all('a')[1]
['href']
30             food_calorie = food.find('p').text
31             writer.writerow([food_name, food_calorie, food_url])

```

```
32         #借助writerow()函数，把提取到的数据：食物名称、食物热量、食物详情链接，写入csv文件。
33         print(food_name)
34
35     csv_file= open('boohee.csv', 'w', newline='')
36     #调用open()函数打开csv文件，传入参数：文件名“boohee.csv”、写入模式“w”、newline=''。
37     writer = csv.writer(csv_file)
38     # 用csv.writer()函数创建一个writer对象。
39     writer.writerow(['食物', '热量', '链接'])
40     #借助writerow()函数往csv文件里写入文字：食物、热量、链接
41
42     tasks_list = []
43     for x in range(5):
44         task = gevent.spawn(crawler)
45         tasks_list.append(task)
46     gevent.joinall(tasks_list)
```