链滴

一次 python 爬虫实践——黑客派签到脚本

作者：denny0207
原文链接：https://ld246.com/article/1564262699573
来源网站：链滴

前言

最近看到猪哥的爬虫教程后，自己还是很想试一试的。也是偶然的契机，发现了黑客派这个论坛。这有了下面这个脚本

```
import requests
# requests V2.21.0
from bs4 import BeautifulSoup as bs
# beautifulsoup4 V4.8.0

# 保存 cookie
session = requests.Session()

# 登录
def login_hacpai():
    login_url = 'https://hacpai.com/login'

    headers = {'User-Agent':'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (K
TML, like Gecko) Chrome/74.0.3729.169 Safari/537.36',
'Referer':'https://hacpai.com/login'}
    raw_data = {"nameOrEmail":"hahahhahah",
 "userPassword":"hahahahahha", "captcha":""}

    try:
        request = session.post(login_url, headers=headers, json=raw_data)
        request.raise_for_status()
    except requests.exceptions.HTTPError as err:
        print('登录失败!','HTTPError: {}'.format(err))
    except Exception as err:
        print('登录失败!', 'error: {}'.format(err))
    else:
```

原文链接：一次 python 爬虫实践——黑客派签到脚本

```
        result = request.text.split(',')
        respons_ls = result[:2]
        result_ls =[]
        for i in respons_ls:
            result_ls.append(i.split(":")[1])
        return result_ls


# 获取签到链接
def  get_url():
    url =  'https://hacpai.com/activity/daily-checkin'

    headers =  {'User-Agent':'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (K
TML, like Gecko) Chrome/74.0.3729.169 Safari/537.36',
'Referer':'https://hacpai.com/'}

    try:
        request = session.get(url,  headers=headers)
        respons = request.text
        class_= "module__body ft__center vditor-reset"
        soup = bs(respons, 'lxml').find('div', class_=class_)
        # sign_url_soup = soup.find_all('a', class_="btn green")
        for i in soup('a'):
            link = i.get('href')
        return link
    except (SyntaxError, ImportError,
            UnicodeEncodeError, AttributeError) as err:
        print('error: {}'.format(err))
    except Exception as err:
        print('请求失败!', 'error: {}'.format(err))


# 签到
def  sign(url):
    headers =  {'User-Agent':'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (K
TML, like Gecko) Chrome/74.0.3729.169 Safari/537.36',
'Referer':'https://hacpai.com/activity/checkin'}

    try:
        request = session.get(url,  headers=headers)
        request.raise_for_status()
    except requests.exceptions.HTTPError as err:
        print('请求失败!','HTTPerror: {}'.format(err))
    except  Exception  as err:
        print('请求失败!',  'error: {}'.format(err))
    else:
        respons = request.text
        soup =  bs(respons,  'lxml').find('div',  class_="vditor-reset")

        for s in soup.div.strings:
            s = s.replace(' ',  '')
            print(s,end=' ')
```

```
def main():
    sign_in_url = 'https://hacpai.com/activity/checkin'
    result_ls = login_hacpai()
    if result_ls[0] == 'false':
        print('登录失败!', result_ls[1], sep='\n')

    else:
        print('登录成功!')
        url = get_url()
        if url == None:
            print('未找到签到链接。')
        elif 'points' in url:
            print('今日你已经签过到了。')
            print('可以点击或复制链接：{} 查看 '.format(sign_in_url))
        else:
            sign(url)


if __name__ == "__main__":
    main()
```

在写这个脚本的过程中遇到了不少问题：

1. 模拟登录

在模拟登录的过程中遇到了第一个问题：

用户名，密码都对，就是无法登录。

```
import requests

session = requests.Session()

login_url = 'https://hacpai.com/login'
headers = {...}
data = {...}

try:
    request = session.post(login_url, headers=headers)
    request.raise_for_status()
except requests.exceptions.HTTPError as err:
    print('登录失败!','error: {}'.format(err))
except Exception as err:
    print('登录失败!', 'error: {}'.format(err))
else:
    print(request.text)

''' 输出结果 '''
{"sc":false,"msg":"用户不存在"}
```

后来参考了@mufengcoding的脚本，发现传入的表单数据是JSON类型的。于是导入json库（注：略号表示没有改动过的代码）

```
import requests
import json
```

```
...
raw_data = {...}
try:
    request = session.post(login_url, headers=headers, json=raw_data)
    request.raise_for_status()
except:
    ...
    ...
```

然后阅读官方文档发现还有更简单的写法

此处除了可以自行对 dict 进行编码，你还可以使用 json 参数直接传递，然后它就会被自动编码。这是 2.4.2 版的新加功能：

```
url = 'https://api.github.com/some/endpoint'
payload = {'some': 'data'}
r = requests.post(url, json=payload)
```

更改成如下代码:

```
import requests
...
headers = {...}
raw_data = {...}

try:
    request = session.post(login_url, headers=headers,
json=raw_data)
   request.raise_for_status()
```

2.获取签到链接

登录成功后，接下来的问题就是获取签到的链接。

在开发者工具里很容易在签到页面里获取到链接。我遇到的问题是如何把链接提取出来。通过观察签链接除了域名，请求的资源路径外还有一个传给服务器的参数。通过关键词在网页源代码上查找，发在script标签内发现了我想要的参数。我想，通过提取script标签里的内容把想要的值拿出来，写了下功能：

```
from bs4 import BeautifulSoup as bs
...
def get_token():
    url = '...'
    headers = {...}
    raw_data = {...}
    request = requests.get(url, headers=headers)
    respons = request.text
    soup = bs(respons, 'lxml')

    for i in soup.find('script').strings:
        print(i)
...
```

但就是无法拿到我想要的值，输不出那个字典，不知是我的逻辑出现了问题，还是说这个想法本身就有问题。

不过还好，我有了第二个想法，直接把链接提取出来，... 好吧，是我想太多

```
...
def get_url():
    ...
    try:
        request = session.get(url, headers=headers)
        respons = request.text
        soup = bs(respons, 'lxml')
        for i in soup.find_all('a', class_="btn green"):
            link = i.get('href')
        return link
    except Exception as err:
        print('请求失败!', 'error: {}'.format(err))
...
```

3.输出签到结果

虽然说在这里获取标签里的值没什么大问题，但我还是要在这感谢我那个麻烦的想法，它让我写这个能的时候省了不少力。

最后

这还不是这个脚本的最终形态，里面还会加其他内容。(可能不是什么重要的内容，我对这个脚本的定是复习学过的知识点)

源码

更新说明

1. 增加友好输出
2. 去除输出的响应和一些不必要的异常信息

参考链接

Requests文档

Beautiful Soup 4.4.0 文档

python3.7 实现自动签到

猪哥爬虫专栏

签到活动更新

```
<details>
<summary></summary>
复习：字符串的操作方法，自定义函数，处理多种异常的不同写法，HTTP，文件的读取，列表,字典加元素 <br />
get:爬虫一般步骤，requests, beautifulsoup 第三方库简单使用
</details>
```