

西工大教务系统成绩爬虫-GradeSpider

高丽

2019.03完成

- [西工大教务系统成绩爬虫-GradeSpider](#)
 - [一、实验目的](#)
 - [二、系统设计思路](#)
 - [（1）实现功能](#)
 - [（2）框架思路](#)
 - [（3）所需环境说明](#)
 - [三、详细设计实现](#)
 - [（1）登录](#)
 - [（2）爬取](#)
 - [（3）应对反爬虫](#)
 - [（4）代码实现](#)
 - [四、实验结果及分析](#)
 - [五、源码地址](#)

一、实验目的

1. 掌握基本的爬虫，能够爬取特定的内容
2. 掌握模拟浏览器登录
3. 掌握GET、POST等请求方式
4. 了解反爬虫

二、系统设计思路

（1）实现功能

1. 模拟登录教务系统
2. 按用户需求爬取相应成绩
3. 一个简单的应对反爬虫的策略

（2）框架思路

1. 首先爬取登录页面的表单中的其他信息，记录下来，输入学号、密码，提交所有表单信息，随机选择User-agent模拟浏览器登录；
2. 添加session，使得能够访问同一个网站的不同页面；
3. 代码本身通用，支持爬取所有学期，不过考虑到2017年入学，之前没有成绩，故只给出从“2017-2018秋”至今所有学期的选项，可以查看单个学期，也可以查看所有学期。输入选项，按需求爬取，保存至 `output/***.xlsx`

（3）所需环境说明

- 需要requests、bs4、(lxml)、pandas、(openpyxl)
- 注意，需将所有库都更新到最新，否则部分函数无法使用：

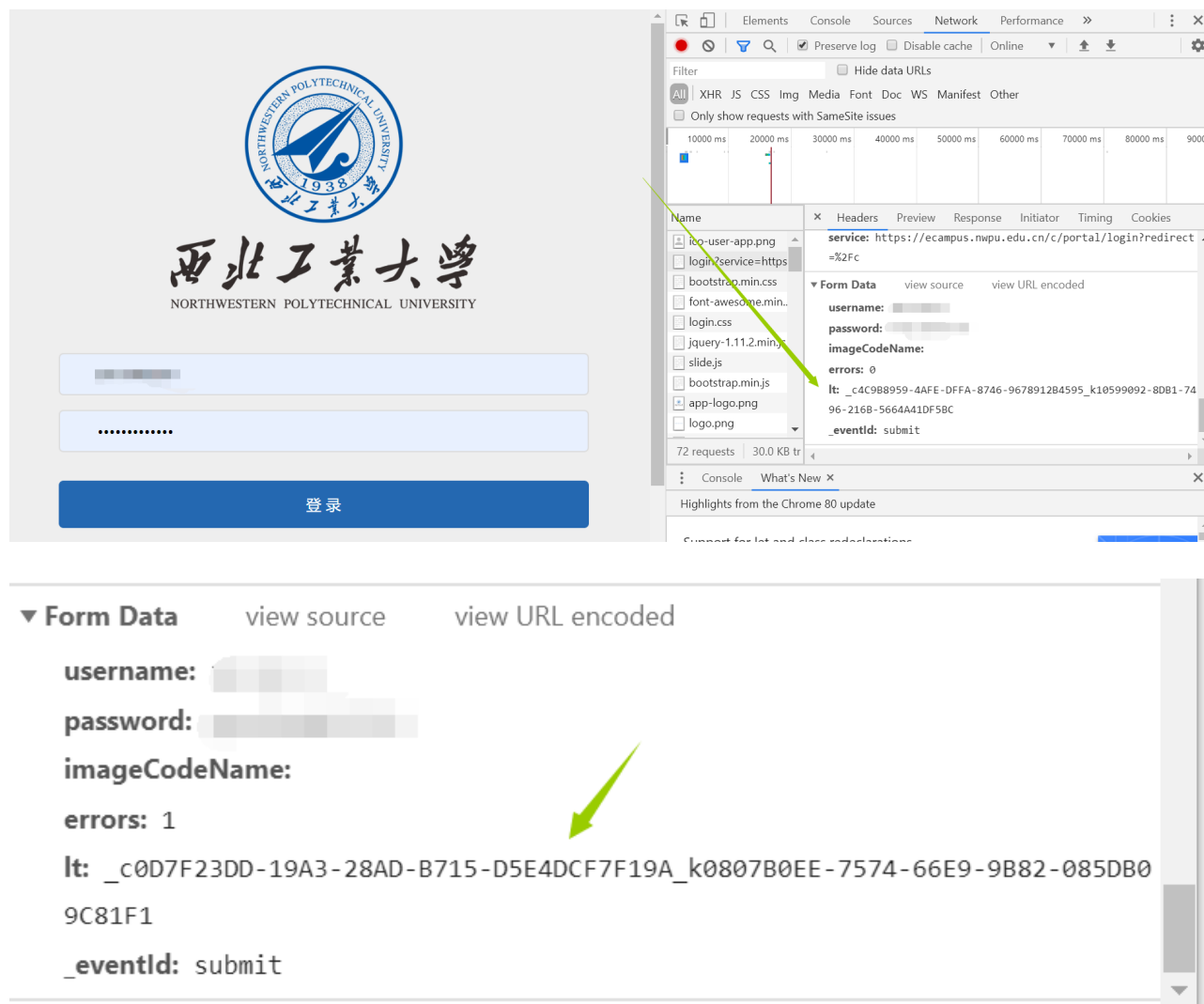
```
pip list # 查看已安装的所有的依赖包
pip list --outdated -- format==columns # 像表格一样列出所有已安装的依赖包的当前版本和可升级版本
# 升级所有依赖包含如下两个命令
pip install pip-review --user # 先安装pip-review函数
pip-review --local --interactive # 成功升级所有的依赖包
```

三、详细设计实现

(1) 登录

- 问题A

一开始决定从翱翔门户登录，再跳转至教务系统，进入成绩模块，但是发现怎么都无法登录成功。于是查询Network-Headers-Form Data发现：



该处绿色箭头表示有一个随机数 **lt** 以及其他信息，随机数每次提交表单前就已经存在，并且每次提交完以后会变化。其他信息是固定值。

同时，还发现学号和密码都是明文表示，也没有验证码和其他特殊信息。

- 解决方法

应对策略是：先GET请求网页，将除了学号、密码以外的所有表单信息全部记录下来，接着连同学号和密码一起提交，即可登录。

- 问题B



成功登录后还是遇到了另外一个问题：即使利用session也无法在翱翔门户跳转到教务系统。

- 解决方法

并没有找到具体原因。

于是放弃从翱翔门户进入教务系统，而是直接找到教务系统的特定登录url：

<http://us.nwpu.edu.cn/eams/login.action> 然后继续观察FormData，发现这次登录的表单信息更加简单，省略了随机数，仅有username、password、encodedPassword(这里为空)、session_locale（设置中英文的选项，中文用“zh-CN”表示）

▼ Form Data	view source	view URL encoded
username:		
password:		
encodedPassword:		
session_locale:	zh_CN	

最后顺利登录。

- 关于判断是否登录成功：

原来的页面没有登录或者登录失败时，源码里有一个input标签，其属性name值为username,所以如果能够找到这个标签，则代表登录失败，如果没有，则代表已经跳转——登录成功。

```
<div class="input-box">
  <input type="text" name="username" id="username" placeholder="请输入用户名" class="login-username">
</div>
<div class="input-box">
  <input type="password" name="password" id="password" placeholder="请输入密码" class="login-password">
  <input name="encodedPassword" type="hidden" value="" />
</div>
```

(2) 爬取

成绩部分的url非常有规律，可以直接指定参数semesterId以访问不同学期的成绩单。例如访问2019-2020秋学期可以设置semesterId为19，其对应网址为

<http://us.nwpu.edu.cn/eams/teach/grade/course/person!search.action?semesterId=19&projectType=>

- 问题

成绩部分的代码虽然并没有用js或其他机制，是可以直接爬取的，但还是遇到了问题——每个学期的成绩表格格式不一样，例如有的学期有实验成绩，而有的学期没有；

我的成绩

所有学期成绩

学年学期: 2019-2020学年秋学期 切换学期

学年学期	课程代码	课程序号	课程名称	课程类别	学分	平时成绩	期中成绩	实验成绩	期末成绩	总评成绩	最终	绩点
2019-2020 秋	U09M13001	U09M13001.01	嵌入式系统及应用 (双)	专业选修课程	4	100	100	91	92	94	94	3.9
2019-2020 秋	U09M11129	U09M11129.01	计算机网络	学科基础课程	3.5	97			96	96	96	4
2019-2020 秋	U09M11131	U09M11131.01	密码学	专业核心课程	3	97			90	93	93	3.9
2019-2020 秋	U31G71001T	U31G71001T.03	体育1 (瑜伽)	体育	1				89	89	89	3.8
2019-2020 秋	U30L11006	U30L11006.01	戏剧鉴赏	综合素养	2	95			96	96	96	4
2019-2020 秋	U09M11182	U09M11182.01	软件安全	专业核心课程	2	82		95		86	86	3.6
2019-2020 秋	U09M11125	U09M11125.01	计算机操作系统	学科基础课程	3.5	90			87	88	88	3.7
2019-2020 秋	U09P61004	U09P61004.01	计算机网络实训	集中实践环节	2			99		99	99	4
2019-2020 秋	U33L11016	U33L11016.01	音乐图像电影文件编辑处理	综合素养	2				98	98	98	4

并且，经过测试发现，不同人在同一学期的表格也不一样，有的人有补考成绩，有的人没有。所以不能用简单的索引来读取。

再加上，源码中这些选项之间在属性和标签上没有任何差异，无法通过BeautifulSoup的find函数读取特定的某一属性：

```
<tbody id="grid16310936501_data"><tr>                <td>2019-2020 秋</td>
                <td>U09M13001</td>
                <td>U09M13001.01</td>
                <td>
                    <a href="javascript:void(0)" onclick="showInfo(37396760)" title="查看成绩详情">嵌入式系统及应用（双）</a>
                </td>
                <td>专业选修课程</td>
                <td>4</td>
                <td style="">100
            </td><td style="">100
            </td><td style="">91
            </td><td style="">92
            </td><td style="">94
            </td><td style="">94
            </td><td style="">3.9
        </td><td>
            <td>2019-2020 秋</td>
```

一节课

解决方法

直接全部读取，然后利用pandas的DataFrame数据结构保存,可以实现项名和值之间的映射,非常方便。

首先利用字典将“课程序号”“课程代码”“课程名称”等这些属性全部读取，接着逐行读取数据（每个tr属性代表一节课，每个td属性代表一项数值），最后转化为DataFrame格式。这样代码对于任何格式的表格都可以直接读取。

并且pandas只需调用to_excel函数即可输入到xlsx文件。

(3) 应对反爬虫

实际上并没有发现教务系统的登录和成绩单有明显的反爬虫的机制（教务处的登录表单上没有随机数，复杂程度与翱翔门户比也差了很多，数据也没有被隐藏）。

此外，甚至也没有看到robots协议。

但是为了以防万一，还是写了一个常用的、比较简单的应对反爬虫的策略——python的默认User-agent会暴露自己爬虫的身份，所以这里构造UA池，每次随机选用百度、谷歌、safari、Maxthon浏览器中的一个User-agent访问，这样可以认为是一个小网络中多个用户同时访问一个页面，即不会被认为是爬虫。

```
ua_list=[#ua池
'Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)',
#百度
'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.94 Safari/537.36',
#谷歌
```

```
'Mozilla/5.0 (Windows; U; Windows NT 6.1; en-us) AppleWebKit/534.50 (KHTML,
like Gecko) Version/5.1 Safari/534.50',
# Safari
'Mozilla/4.0(compatible;MSIE7.0;WindowsNT5.1;Maxthon2.0)'
# (Maxthon)
]
```

(4) 代码实现

```
import requests
from bs4 import BeautifulSoup as bs
import os
import random
import re
import pandas as pd
import sys

path = os.path.abspath(os.path.dirname(sys.argv[0]))#获得当前路径
outputpath = path+'\\output'#创建output文件夹用

base_url='http://us.nwpu.edu.cn/eams/login.action'#登录页面
def mkdir(path):#创建文件夹
    path =path.strip()#去除空符合
    path = path.rstrip("\\") # 去除尾部 \ 符号
    isExists = os.path.exists(path)
    if not isExists:
        # 如果不存在则创建目录
        # 创建目录操作函数
        os.makedirs(path)
        return True
    else:
        # 如果目录存在则不创建
        return None

def login(s,username,password):
    #登录, 随机选择一个User-agent模拟浏览器登录, 一定程度上可以应对反爬虫
    ua_list=[#ua池
        'Mozilla/5.0 (compatible; Baiduspider/2.0;
+http://www.baidu.com/search/spider.html)',
        #百度
        'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/50.0.2661.94 Safari/537.36',
        #谷歌
        'Mozilla/5.0 (Windows; U; Windows NT 6.1; en-us) AppleWebKit/534.50
(KHTML, like Gecko) Version/5.1 Safari/534.50',
        # Safari
        'Mozilla/4.0(compatible;MSIE7.0;WindowsNT5.1;Maxthon2.0)'
        #傲游 (Maxthon)
    ]
    ua = random.choice(ua_list)#随机选择一个ua
```

```

print("模拟浏览器, 随机选择User-agent: "+ua)
# 首先获取到登录界面的html# 教务系统
html = s.get(base_url, headers={'User-agent': ua})
soup = bs(html.text, 'lxml')
# 找到form的验证参数
encodedPassword = soup.find('input', attrs={'name': 'encodedPassword'})
['value']
session_locale = soup.find('input', attrs={'name': 'session_locale'})['value']

# 构造需要post的参数表
FormData = {
    'username': username,
    'password': password,
    'encodedPassword': encodedPassword,
    'session_locale': session_locale,
}

print('登录中.....\n.....\n')
# 测试看看是否能找到登陆后的信息 #获得用户信息
post_login = s.post(base_url, data=FormData, headers={'User-agent': ua})
html_login = post_login.text
soup_login = bs(html_login, 'lxml')
try:
    #是否登录成功, 原来的页面没有登录或者登录失败时, 有一个input标签, 属性name值为
    username
    #如果能够找到name代表登录失败, 如果没有, 则成功
    name = soup_login.find('input', attrs={'name': 'username'})
    if name == None:
        print('=====成功登录西北工业大学教务系统=====\n')
        return s
    else:
        print("登录失败, 用户名或密码错误.....")
except: #出现异常
    print("登录失败, 请再试.....")

return None

def search_part(s, termID):
    #根据输入选项查找, 查找完以后返回查找结果
    term_url =
    'http://us.nwpu.edu.cn/eams/teach/grade/course/person!search.action?
    semesterId='+str(termID)+'&projectType=' #特定学期页的url地址
    html_grade = s.get(term_url).text
    soup_grade = bs(html_grade, 'lxml')
    head_grade = soup_grade.find('thead', attrs={'class': "gridhead"}) #找成绩单部分
    heads = head_grade.find_all('th') #找成绩单头部信息
    trs_dict={}
    num=0
    for eachhead in heads:
        #num为index
        trs_dict[num]=[eachhead.text.strip()] #去空格
        num = num +1
    #num为列数
    #找到表示成绩的部分

```

```
tbody = soup_grade.find('tbody')#attrs={'id':'grid16527563961_data'})
#获得表格所有内容
trs = tbody.find_all('tr')

for eachtr in trs:#每一个tr 即每一门课
    tds=eachtr.find_all('td')#获得td的列表
    #放入字典中
    k = 0
    for eachtd in tds:
        trs_dict[k].append(eachtd.text.strip())
        k=k+1

newtrs_dict={}
#重新创建不包含123index的字典, 方便后续放入dataframe中
for i in range(num):
    newtrs_dict[trs_dict[i][0]]=trs_dict[i][1:]

gradetable = pd.DataFrame(newtrs_dict)#转化为dataframe
print(gradetable)#打印
return gradetable

def search_grade(s):
#查询成绩 学期表 存储跳转的url参数semid
term_url_list = [
    ['2017-2018年秋学期',17],
    ['2017-2018年春学期',35],
    ['2018-2019年秋学期',18],
    ['2018-2019年春学期',36],
    ['2019-2020年秋学期',19],
    ['查询以上所有学期',0],
]

print('目前可以查询的学期有——')
#输出可以查找的范围
for i in range(len(term_url_list)):
    print(str(i+1)+'、'+term_url_list[i][0])

inputID = input("输入对应序号查询该学期的成绩: ")
try:
    inputID = int(inputID)
except:
    print('参数错误! ')
    return None
print('\n查询'+term_url_list[inputID-1][0]+'成绩结果:\n')
#成绩单的url以及url参数
if inputID==6:#查找所有学期
    mkdir(outputpath)#创建output文件夹
    writer = pd.ExcelWriter(outputpath+'\\所有学期成绩单.xlsx')#用该函数追加输出
    for i in range(1,6):
        termID=term_url_list[i-1][1]#获得学期的semid参数
        gradetable=search_part(s,termID=termID)#查找函数
        gradetable.to_excel(writer, str(term_url_list[i-1][0]))#输出
        writer.save()
    else:#查找单个学期
```

```

termID=term_url_list[inputID-1][1]
gradetable=search_part(s,termID=termID)
mkdir(outputpath)
gradetable.to_excel(outputpath+'\\{}成绩单
单.xlsx'.format(term_url_list[inputID-1][0]),encoding='utf-8', index=True,
header=True)

print('\n=====成绩单以excel文件格式保存至output文件夹中=====\n')

if __name__ == "__main__":
    while(1):
        username=input("请输入学号/工号: ")
        password = input("请输入密码: ")
        s = requests.session()#设置session, 页面跳转时不会退出
        s= login(s,username,password)#登录
        if s:#登录成功
            while(1):
                search_grade(s)#查找
                cont = input('是否继续? (y or n) ')
                if cont=='y':
                    continue
                else:
                    break
            break
        else:#登录失败
            cont = input('是否继续? (y or n) ')
            if cont=='y':
                continue
            else:
                break
    print('=====END=====\n')

```

四、实验结果及分析

1. 登录：输入学号、密码，得到开始界面：

```

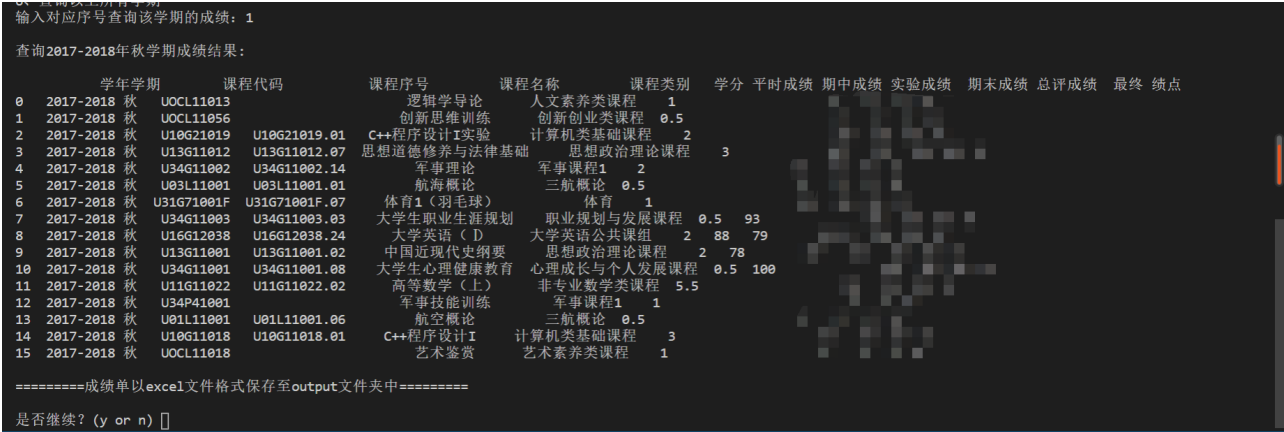
请输入学号/工号: 
请输入密码: 
模拟浏览器，随机选择User-agent: Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)
登录中.....

=====成功登录西北工业大学教务系统=====

目前可以查询的学期有——
1、2017-2018年秋学期
2、2017-2018年春学期
3、2018-2019年秋学期
4、2018-2019年春学期
5、2019-2020年秋学期
6、查询以上所有学期
输入对应序号查询该学期的成绩: 1

```


2. 选择想查看的特定学期或全部学期，这里选择**2017-2018秋学期**，输入**1**:



此外还可以选择继续**y**，输入其他选项查看；如果选择**n**结束循环。

3. 在同级目录下会出现一个新的output文件夹，所有爬虫内容均以excel表格形式保存在这里:

	2017-2018年春学期成绩单.xlsx	2020-03-14 17:41	Microsoft Excel 工...	7 KB
	2017-2018年秋学期成绩单.xlsx	2020-03-15 1:06	Microsoft Excel 工...	7 KB
	2018-2019年秋学期成绩单.xlsx	2020-03-14 16:48	Microsoft Excel 工...	6 KB
	所有学期成绩单.xlsx	2020-03-14 17:40	Microsoft Excel 工...	14 KB

4. 查看其中一个excel文件:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		学年学期	课程代码	课程序号	课程名称	课程类别	学分	平时成绩	期中成绩	实验成绩	期末成绩	总评成绩	最终	绩点
2	0	2018-2019	U09M1113	U09M1113	信息安全	专业选修	2	100		100	80	86	86	3.6
3	1	2018-2019	U08M2101	U08M2101	数字电子	学科基础	1				84	84	84	3.5
4	2	2018-2019	UOCL11022		美学原理	艺术素养	1					P	P	0
5	3	2018-2019	UOCL11006		中国古代	综合素养	2					P	P	0
6	4	2018-2019	U09P4100	U09P4100	认识实习	集中实践	1				90	90	90	3.8
7	5	2018-2019	U32P4100	U32P4100	金工实习	集中实践	2				90	90	90	3.8
8	6	2018-2019	U09M1112	U09M1112	数据库原	学科基础	3	92		97	96.5	96	96	4
9	7	2018-2019	U09M1109	U09M1109	数据结构	学科基础	4	90		96	89	92	92	3.9
10	8	2018-2019	U16G1204	U16G1204	实用英语	大学英语	2	85	87		93	90	90	3.8
11	9	2018-2019	U09P2100	U09P2100	算法设计	集中实践	2			95		95	95	4
12	10	2018-2019	U08M1105	U08M1105	数字电子	学科基础	3.5	99			95.5	96	96	4
13	11	2018-2019	U13G1100	U13G1100	马克思主	思想政治	3	95		89	80	86	86	3.6

和翱翔门户的成绩单完全符合。

5. 性能指标:

- 定义性能指标: **爬虫结果的准确性**，为了验证，找了其他同学帮忙测试，均未出错。在该指标上表现优异。
- 此外，由于数据量并不大，并且输出发现时间上并无明显差异，故不考虑**时间**指标。

五、源码地址

<https://github.com/cimeguy/GradeSpider>