Python网络爬虫笔记（四）：使用selenium获取动态加载的内容

（一）  说明

上一篇只能下载一页的数据，第2、3、4....100页的数据没法获取，在上一篇的基础上修改了下，使用selenium去获取所有页的href属性值。

使用selenium去模拟浏览器有点麻烦，例如有300页就要点300次（按博客园这种一页20条，也就是6000条数据。要是几百万条，这个就得点好长时间了）

研究下有没有办法调用JS修改页面默认显示的数据条数（例如：博客园默认1页显示20条，改成默认显示1万条数据）。

（二）  完整代码

delayed.py的代码还是和之前一样。最好限速，不限速很容易被拒绝连接，而且也不道德。

（三）结果

下面这个异常是，有的随笔上传了微信公众号的图片（暂时不确定是全部这样，还是部分这样），解析这个的时候会出现编码错误，目前的处理是输出异常信息，跳过这张图片。

1 from selenium import webdriver  
 2 from selenium.webdriver.support.ui import WebDriverWait  
 3 from selenium.webdriver.support import expected\_conditions  
 4 from selenium.webdriver.common.by import By  
 5 from selenium.webdriver.support.expected\_conditions import StaleElementReferenceException  
 6 import time  
 7 import urllib.request as ure  
 8 from delayed import WaitFor  
 9 import lxml.html  
 10 import os  
 11 import docx  
 12 #使用selenium获取所有随笔href属性的值，url只能传小类的，例如https://www.cnblogs.com/cate/python/  
 13 def selenium\_links(url):  
 14 driver = webdriver.Chrome()  
 15 driver.maximize\_window()  
 16 driver.get(url)  
 17 #获取最大页数  
 18 maxPage = driver.find\_element\_by\_css\_selector('#paging\_block div.pager a:nth-last-of-type(2)').get\_attribute('text')  
 19 x=1  
 20 url\_list=[]  
 21 #循环获取当前小类所有页面的href  
 22 while x<=int(maxPage):  
 23 time.sleep(1) #隐式 显式等待都尝试了，还是报错，只能等待1秒了(调试又正常运行)  
 24 x +=1  
 25 #等待 Next出现并返回 ,就是博客园翻到下一页的那个元素  
 26 lastPage = WebDriverWait(driver, 30).until(expected\_conditions.element\_to\_be\_clickable((By.CSS\_SELECTOR, '#paging\_block div.pager a:last-child')))  
 27 #等待元素出现并返回list，这里定位的是页面上的随笔  
 28 html = WebDriverWait(driver, 30).until(expected\_conditions.presence\_of\_all\_elements\_located((By.CSS\_SELECTOR, 'a.titlelnk')))  
 29 #迭代，将href属性的值添加到url\_list  
 30 for h in html:  
 31 url\_list.append(h.get\_attribute('href'))  
 32 lastPage.click()  
 33 driver.quit()  
 34 #以列表形式返回所有URL  
 35 return url\_list  
 36 #传入包含url的列表  
 37 def link\_crawler(seed\_url):  
 38 html\_list = []  
 39 #下载crawl\_queue中的所有网页  
 40 waitFor = WaitFor(2)  
 41 x =1  
 42 while seed\_url:  
 43 print('第'+str(x)+'个')  
 44 x+=1  
 45 url = seed\_url.pop()  
 46 #下载限速  
 47 waitFor.wait(url)  
 48 html = download(url)  
 49 html\_list.append(html)  
 50 return html\_list  
 51 #下载网页并返回  
 52 def download(url,user\_agent='FireDrich',num=2):  
 53 print('下载:'+url)  
 54 #设置用户代理  
 55 headers = {'user\_agent':user\_agent}  
 56 request = ure.Request(url,headers=headers)  
 57 try:  
 58 #下载网页  
 59 html = ure.urlopen(request).read()  
 60 except ure.URLError as e:  
 61 print('下载失败'+e.reason)  
 62 html=None  
 63 if num>0:  
 64 #遇到5XX错误时，递归调用自身重试下载，最多重复2次  
 65 if hasattr(e,'code') and 500<=e.code<600:  
 66 return download(url,num-1)  
 67 return html  
 68   
 69 #将下载的网页写入Word文档  
 70 def createWord(html):  
 71 x = 0  
 72 while html:  
 73 url = html.pop()  
 74 tree = lxml.html.fromstring(url) # 解析HTML为统一的格式  
 75 title = tree.xpath('//a[@id="cb\_post\_title\_url"]') # 获取标题  
 76 the\_file = tree.xpath('//div[@id="cnblogs\_post\_body"]/p') # 获取正文内容  
 77 pre = tree.xpath('//pre') # 获取随笔代码部分（使用博客园自带插入代码功能插入的）  
 78 img = tree.xpath('//div[@id="cnblogs\_post\_body"]/p/img/@src') # 获取图片  
 79 # 修改工作目录  
 80 os.chdir('F:\Python\worm\data\博客园文件')  
 81 try:  
 82 # 创建一个空白新的Word文档  
 83 doc = docx.Document()  
 84 # 添加标题  
 85 doc.add\_heading(title[0].text\_content(), 0)  
 86 #有的设置成注册用户才能浏览的随笔，调用download函数时下载不到正确的网页，导致获取不到标题  
 87 #title会是空列表，这里忽略这篇随笔,利用http.cookiejar模块应该可以解决这种问题，以后再看看这个模块了  
 88 except IndexError as e:  
 89 continue  
 90 for i in the\_file:  
 91 # 将每一段的内容添加到Word文档（p标签的内容）  
 92 doc.add\_paragraph(i.text\_content())  
 93 # 将代码部分添加到文档中  
 94 for p in pre:  
 95 doc.add\_paragraph(p.text\_content())  
 96 # 将图片添加到Word文档中  
 97 for i in img:  
 98 try:  
 99 ure.urlretrieve(i, '0.jpg')  
100 doc.add\_picture('0.jpg')  
101 except UnicodeEncodeError as e:  
102 print('异常'+str(e))  
103 pass  
104 # 截取标题的前8位作为Word文件名  
105 filename = title[0].text\_content()[:8] + '.docx'  
106 # 保存Word文档  
107 # 如果文件名已经存在，将文件名设置为title[0].text\_content()[:8]+ str(x).docx，否则将文件名设置为filename  
108 if str(filename) in os.listdir('F:\Python\worm\data\博客园文件'):  
109 doc.save(title[0].text\_content()[:8] + str(x) + '.docx')  
110 x += 1  
111 else:  
112 doc.save(filename)  
113 #调用selenium\_links获取所有url  
114 html = selenium\_links('https://www.cnblogs.com/cate/ruby/')  
115 #调用link\_crawler下载所有网页  
116 downHtml = link\_crawler(html)  
117 #提取已经下载的网页数据到Word文档中  
118 createWord(downHtml)

