MAZ-CAS-DDJ / **kurs_18_19**

Branch: master ▾ **kurs_18_19** / 10 Selenium, Projekte / **02 Arbeit mit Selenium.ipynb** Find file Copy path

barjacks 10 cfb3331 on 9 Aug

1 contributor

369 lines (368 sloc) 11 KB

Arbeit mit Selenium

Die Arbeit mit Selenium erfordert etwas Übung. Aber der Zeitaufwand lohnt sich. Es gibt mit Selenium kaum ein Webdienst der nicht scrapbar wird. Beginnen wir aber wie üblich mit der Dokumentation. Sie ist im Falle von Selenium sehr hilfreich. Ihr findet sie hier (http://selenium-python.readthedocs.io/). Und hier (http://selenium-python.readthedocs.io/locating-elements.html).

Um Selenium kennenzulernen, gehen wir zurück zu unserem Beispiel der Lehren: https://www.berufsberatung.ch/dyn/show/2930 (https://www.berufsberatung.ch/dyn/show/2930). Nun wollen wir keine URLs generieren, um unsere Inhalte zu finden. Wir wollen stattdessen mit der Site interagieren. So sollten wir alle Einträge bekommen. BeautifulSoup werden wir trotzdem noch dazu nehmen. Denn Selenium liest keine Inhalte aus. Die Library lässt uns einfach mit dem Webdienst interagieren.

Beginnen wir mit den Imports

In [ ]:
```
from bs4 import BeautifulSoup
import requests
import time
import datetime
import pandas as pd
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
```

Und dann schicken wir Selenium auf die Seite.

In [ ]:
```
#Wir starten den Browser:
driver = webdriver.Chrome('/usr/local/bin/chromedriver')
#Wir besuchen die Site
driver.get("https://www.berufsberatung.ch/dyn/show/2930")
```

Nun suchen wir mit dem Inspector die Elemente, die wir ansteuern wollen.

In [ ]:
```
driver.find_element_by_class_name("fs-autocomplete-trigger").click()
```

In [ ]:
```
driver.find_element_by_id("sw_453").click()
```

Aber wir wollen alle Ergänzen. Holen wir deshalb die Nummern nochmals

In [ ]:
```
r = requests.get('https://www.berufsberatung.ch/dyn/show/2930')
soup = BeautifulSoup(r.text, 'lxml')
ids = []
for elem in soup.find('ul',{'class':'ui-autocomplete full-list '}).find_all('a'):
    elem = "sw_" + elem['data-id']
    ids.append(elem)
```

Testen wir es mit den ersten fünf Einträgen

In [ ]:
```
for elem in ids[:4]:
    print(elem)
    time.sleep(.5) #damit es nicht zu schnell geht
    driver.find_element_by_class_name("fs-autocomplete-trigger").click()
    time.sleep(.5)
    driver.find_element_by_id(elem).click()
```

In [ ]:
```
driver.find_element_by_id("uxfs-action").click()
```

Zeigen wird alle Ergebnisse an

In [ ]:
```
driver.find_element_by_id("aSearchPaging").click()
```

Speichern wir die Ergebnisse ab

In [ ]:
```
text = driver.page_source
```

In [ ]:
```
def lehrstellen(html):

    soup = BeautifulSoup(html, 'lxml')

    ortsliste = soup.find('div', {'class':'resultpart result-body'})\
                    .find_all('div', {'class':'display-table-cell table-col-3'})

    firmenliste = soup.find('div', {'class':'resultpart result-body'})\
```

```
                    .find_all( div , { class : display-table-cell bold company data-id table-col-
1'})

    jahresliste = soup.find('div', {'class':'resultpart result-body'})\
                    .find_all('div', {'class':'display-table-cell float-left-for-sd table-col-4'})

    anzahlliste = soup.find('div', {'class':'resultpart result-body'})\
                    .find_all('div', {'class':'display-table-cell text-align-center float-left-for
-sd table-col-5'})

    lst = []

    for ort, firma, jahr, anzahl in zip(ortsliste,firmenliste,jahresliste, anzahlliste):

        mini_dict = {'Ort':ort.text,
                     'Firma':firma.text,
                     'Jahr':jahr.text,
                     'Anzahl':int(anzahl.text.replace(' Lehrstelle(n)\n','').replace('\n',''))}
        lst.append(mini_dict)

    return lst
```

In [ ]:
```
lehrstellen(text)
```

Bringen wir alles zusammen

In [ ]:
```
#Funktion, um nur die Informationen herauszuziehen, die uns interessieren
def lehrstellen(html):

    soup = BeautifulSoup(html, 'lxml')
    try:
        ortsliste = soup.find('div', {'class':'resultpart result-body'})\
                    .find_all('div', {'class':'display-table-cell table-col-3'})

        firmenliste = soup.find('div', {'class':'resultpart result-body'})\
                    .find_all('div', {'class':'display-table-cell bold company data-id table-col-
1'})

        jahresliste = soup.find('div', {'class':'resultpart result-body'})\
                    .find_all('div', {'class':'display-table-cell float-left-for-sd table-col-4'})

        anzahlliste = soup.find('div', {'class':'resultpart result-body'})\
                    .find_all('div', {'class':'display-table-cell text-align-center float-left-for
-sd table-col-5'})

        lehrstelle = soup.find('ul',{'class':'ui-autocomplete full-list '})\
                    .find_all('a')
        lst = []

        for ort, firma, jahr, anzahl,lehr in zip(ortsliste,firmenliste,jahresliste, anzahlliste,le
hrstelle):

            mini_dict = {'Ort':ort.text,
                     'Firma':firma.text,
                     'Jahr':jahr.text,
                     'Anzahl':int(anzahl.text.replace(' Lehrstelle(n)\n','').replace('\n','')),
                     'Lehrstelle':lehr['data-value']}
            lst.append(mini_dict)

        return pd.DataFrame(lst).to_csv("d/"+str(datetime.datetime.now())+".csv")

    except:
        return pd.DataFrame([{'Ort':'Keine Treffer',
                'Firma':'Keine Treffer',
                'Jahr':'Keine Treffer',
                'Anzahl':'Keine Treffer'}])

#Bauen wir Listen aller Job-IDs
r = requests.get('https://www.berufsberatung.ch/dyn/show/2930')
soup = BeautifulSoup(r.text, 'lxml')
ids = []
for elem in soup.find('ul',{'class':'ui-autocomplete full-list '}).find_all('a'):
    elem = "sw_" + elem['data-id']
    ids.append(elem)

#Teilen wir diese Listen mit Länge von je 5 Teilen.
#Das habe ich nicht selber geschrieben, sondern hier geholt:
#https://stackoverflow.com/questions/312443/how-do-you-split-a-list-into-evenly-sized-chunks
idslst = [ids[i:i + 5] for i in range(0, len(ids), 5)]

count = 0
for ids in idslst:

    #Starten wir den Chrome-Browser und besuchen die Site
```

```
driver = webdriver.Chrome('/usr/local/bin/chromedriver')
driver.get("https://www.berufsberatung.ch/dyn/show/2930")

#Bereiten wir die Suche vor
for elem in ids:
    time.sleep(1) #damit es nicht zu schnell geht
    driver.find_element_by_class_name("fs-autocomplete-trigger").click()
    time.sleep(1)
    driver.find_element_by_id(elem).click()

#Suchen wir
time.sleep(1)
driver.find_element_by_id("uxfs-action").click()

#Nun nun sorgen wir dafür, dass alle Ergebnisse anzeigt werden.
exists = 1
while(exists==1):
    loadmore = driver.find_element_by_id("aSearchPaging")
    if loadmore.text == "MEHR ERGEBNISSE ANZEIGEN":
        driver.find_element_by_id("aSearchPaging").click()
        time.sleep(1)
    else:
        exists = 0

print(count)
count += 1
```