

15/08/2019

# Selenium ve BeautifulSoup ile Web Scrapping

TUBİTAK BİLGEM  
b3lab



BULUT BİLİŞİM VE BÜYÜK VERİ  
ARAŞTIRMA LABORATUVARI

CLOUD COMPUTING & BIG DATA  
RESEARCH LABORATORY

Yasin Bursalı

yasinbursali38@gmail.com  
/in/yasinbursali/

# NEDİR?

Web scraping, web sitelerinden çeşitli araçlar yardımıyla veri çekmektir. Web scraping yazılımı World Wide Web'e HTP protokolünü kullanarak direkt erişebileceği gibi, tarayıcı yoluyla da erişebilir.

# AMAÇ

Bu projenin amacı, BeautifulSoup ve Selenium paketlerini kullanarak CarrefourSA Market üzerinden ürün, kategori, fiyat verilerini çekmek, ve bu süreci otomatize hale getirmektir.

# AŞAMALAR

01

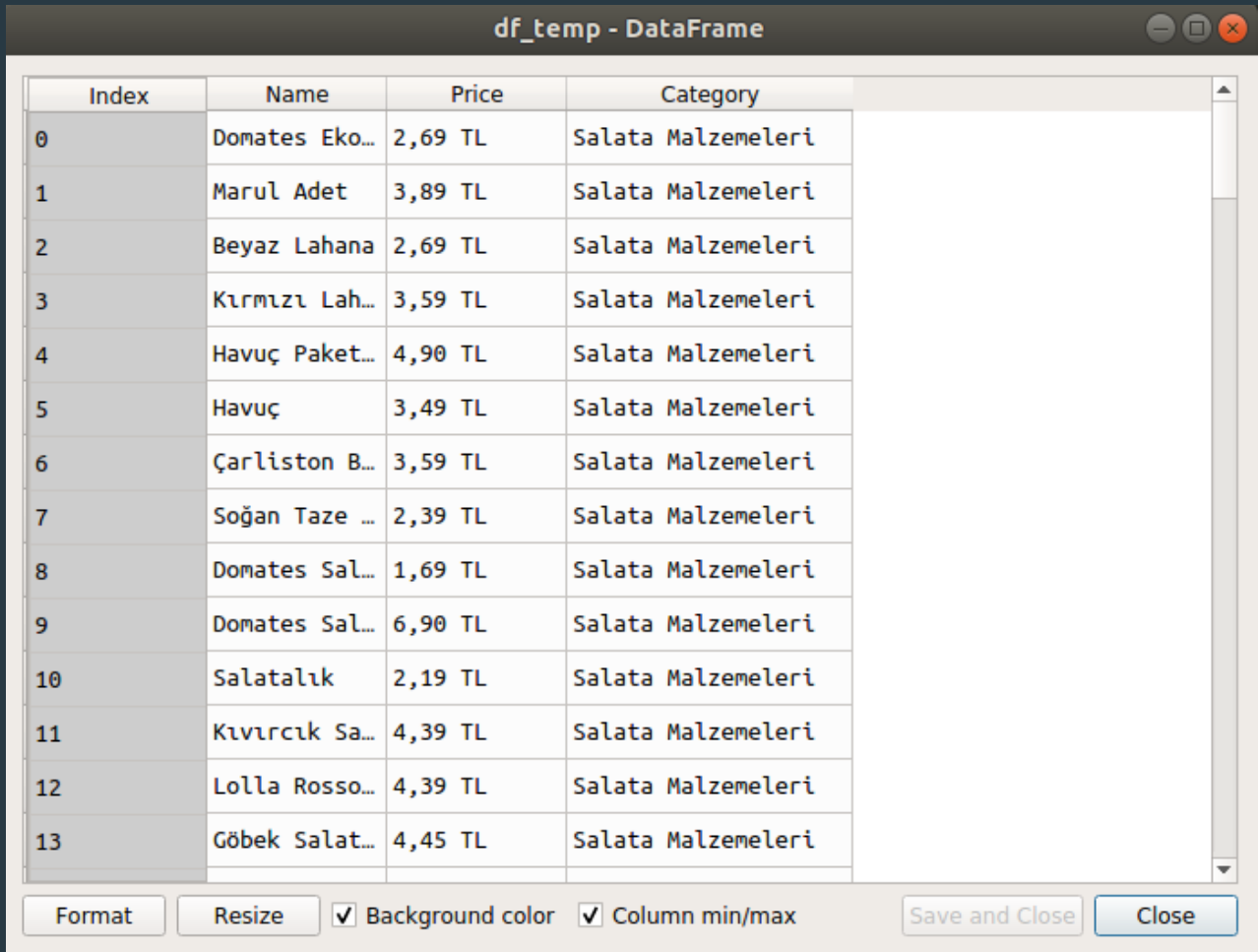
İnternet üzerinden BeautifulSoup paketini kullanarak yapılan scrapping projeleri incelendi.

bs4 -> Scrapping işlemi  
requests -> Web siteleri ile iletişim  
sys -> recursionlimit() fonksiyonu  
pandas -> Veriyi tutmak için

Paketlerinin proje için gerekli olduğuna karar verildi.

# 02

Başlangıç olarak herhangi bir sorun olmadan, bs4 paketi kullanılarak tek bir sayfadan veri çekilip bir pandas dataframe'inde tutuldu.



df\_temp - DataFrame

Index	Name	Price	Category
0	Domates Eko...	2,69 TL	Salata Malzemeleri
1	Marul Adet	3,89 TL	Salata Malzemeleri
2	Beyaz Lahana	2,69 TL	Salata Malzemeleri
3	Kırmızı Lah...	3,59 TL	Salata Malzemeleri
4	Havuç Paket...	4,90 TL	Salata Malzemeleri
5	Havuç	3,49 TL	Salata Malzemeleri
6	Çarliston B...	3,59 TL	Salata Malzemeleri
7	Soğan Taze ...	2,39 TL	Salata Malzemeleri
8	Domates Sal...	1,69 TL	Salata Malzemeleri
9	Domates Sal...	6,90 TL	Salata Malzemeleri
10	Salatalık	2,19 TL	Salata Malzemeleri
11	Kıvırcık Sa...	4,39 TL	Salata Malzemeleri
12	Lolla Rosso...	4,39 TL	Salata Malzemeleri
13	Göbek Salat...	4,45 TL	Salata Malzemeleri

Format Resize ☒ Background color ☒ Column min/max Save and Close Close

# 03

Aynı kategoride, bir sonraki sayfaya giderek oradaki verileri de çekecek ve sonrasında dataframe'e append edecek şekilde veri çekilmek istendi. İlk başta sadece bs4 paketinin kullanılması düşünülürken, bs4'ün tek başına yeterli olmadığı, sayfadaki butonlara tıklama gibi interaktif işlemler söz konusu olduğunda yetersiz kaldığı fark edildi. Bu noktadan sonra selenium paketinin de projeye dahil edilmesi kararlaştırıldı. Ayrıca önceki projelerin raporları incelendi.

# 04

Selenium paketi'nin webdriver modülü kullanıldı. Bunun için chrome driver indirildi ve gerekli ayarlamalar yapıldı.

# 05

URL adresini parametre olarak alan ve o sayfayı parse ettikten sonra bir ileri sayfaya tıklayan, sonrasında bir dataframe döndüren parsePage() fonksiyonu tanımlandı.

Muhtemelen, selenium belirtilen XPATH'e uygun olan ama istenmeyen elementleri de yakaladığından, NoSuchElementException alındı. Try-Except yapısı ile çözüldü. Selenium'la alakalı buna benzer sorunlar çok uzun vakit aldı.

```
def parsePage(currentURL):  
  
    page_df = pd.DataFrame({'Name' : [], 'Price' : [], 'Category' : []}, index=None)  
  
    while(True):  
  
        driver.get(currentURL)  
  
        try:  
            #Getting data for each page  
            r = requests.get(currentURL)  
            soup = bs4.BeautifulSoup(r.text, "xml")  
  
            #Defining parse functions. These functions are taking indexes for finding all elements for giving order.  
            #They return data that pulled from site as string.  
            def parseName(index):  
  
                name = soup.find_all('span', {'class': 'item-name'})[index].text  
                return name  
  
            def parsePrice(index):  
  
                price = soup.find_all('span', {'class': 'item-price'})[index].text  
                return price  
  
            def parseCategory(index):  
  
                category = soup.select('input[name=productMainCategoryPost]')[index]['value']  
                return category  
  
            for i in range(len(soup.find_all('span', {'class': 'item-name'}))):  
                page_df = page_df.append({'Name' : parseName(i), 'Price' : parsePrice(i), 'Category' : parseCategory(i)}, ignore_index=True)  
  
            #Clicking button to navigate next page  
            button = driver.find_element_by_xpath("//a[@class='pr-next']")  
            sleep(1.5)  
            button.click()  
            #For remembering last url  
            currentURL = driver.current_url  
            |  
        except NoSuchElementException:  
            break  
  
    return page_df
```

## 06

İlk önce, bir kategori bittikten sonra anasayfaya dönüp sıradaki kategoriye tıklamak gibi bir yol izlendi. Ama seleniumdaki bazı sorunlardan dolayı anasayfaya döndükten sonra bir sıradaki kategoriye tıklanamadı.

NoSuchElementException veya StaleElementReferenceException hatası alındı. Bunun selenium paketinde kronik bir sorun olduğu tespit edildi.

Work-around olarak, daha önceden sayfadaki bütün kategori sayfalarının bir listede tutulmasına karar verildi.

# 07

Burada da sayfa linkleri;  
[carrefoursa.com/tr/meyve-sebze/c/1014](https://carrefoursa.com/tr/meyve-sebze/c/1014)  
[carrefoursa.com/tr/meyve/c/1015](https://carrefoursa.com/tr/meyve/c/1015)  
şeklinde, alt-üst kategoriler düzensiz olduğundan, (xpath'leri de aynı şekilde ayırt edilemiyor, class isimleri hepsinin aynı.) bütün ürünler iki defa listeleniyor ve bu istenmeyen bir durum.





# 08

Bu sorunu daha iyi yönetebilmek için, bütün dataframe'leri üst üste ekleyip tek bir csv dosyası çıkarmak yerine her kategorinin bir dataframe'i ve sonrasında csv dosyası çıkması sağlandı.

Program, "Output\_" + currentTime isimli bir dosya oluşturuyor, böylece program her periyodik olarak çalıştığında benzersiz bir isim oluşmuş oluyor, dolayısıyla üzerine yazmıyor. Program sonrasında ilgili csv dosyalarını Output dosyasının içine atıyor.

# 09

scheduler\_carrefour\_scrap.py dosyası ile programın hangi periyotta çalışması gerektiği ayarlanabilir.

# Kaynakç

## a

- <https://docs.seleniumhq.org/docs>
- <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- <https://www.geeksforgeeks.org/python-schedule-library/>
- <https://www.youtube.com/watch?v=rONhdonaWUo&t=301s>
- <https://selenium-python.readthedocs.io/>