Web Scraping Using Python: A Step By Step Guide

Article · September 2019

CITATIONS
3

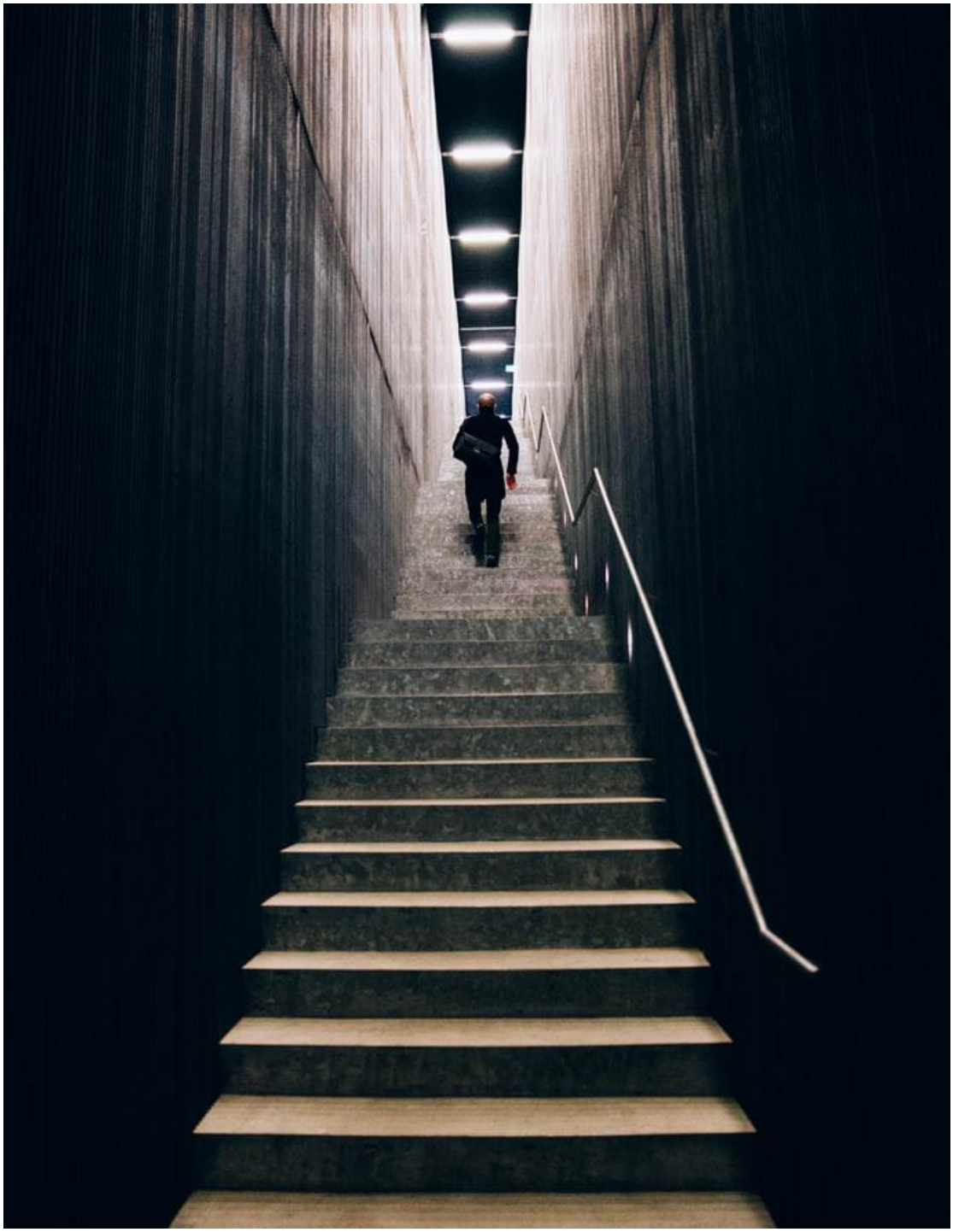
READS
19,823

1 author:

Jiahao Wu
Fordham University
1 PUBLICATION **3** CITATIONS

SEE PROFILE

web scraping using python: a step by step guide



Photoed by Heidi Sandstrom on Unsplash

The need of extracting data from website is increasing. When we are conducting data related projects such as price monitoring, business analytics or news aggregator, we would always need to record the data from website. However, copying and pasting data line by line has been outdated. In this article, we would teach you how to become an “insider” in extracting data from website, which is to do web scraping with python.

Step 0: Introduction

Web scraping is a technique which could help us transform HTML unstructured data into structed data in spreadsheet or database. Besides using python to write cods, there are many other ways to web scraping such as accessing website data with API or data extraction tools like Octoparse.

For some big websites like Airbnb or Twitter, they would provide API for developers to access their data. API stands for Application Programming Interface, which is an access for two applications to communicate with each other. For most people, API is the most optimal approach to obtain data provided from the website themselves.

However, most websites don't have API service. Sometimes even if they provide API, the data you could get is not what you want. Therefore, writing a python script to build web crawler becomes another powerful and flexible solution.

So why should we use python instead of other languages?

Flexibility: As we know, websites update quickly. Not only the content but also the web structure would change frequently. Python is an easy-to-use language because it is dynamically inputable and highly productive. Therefore, people could change their code easily and keep up with the speed of web updates.

Powerful: Python has a large collection of mature libraries. For example, requests, beautifulsoup4 could help us fetch URLs and pull out information from web pages. Selenium could help us avoid some anti-scraping techniques by giving web crawlers the ability to mimic human browsing behaviors. In addition, re, numpy and pandas could help us clean and process the data.

Step 1: Import Python library

In this tutorial, we would show you how to scrape reviews from Yelp. We will use two libraries: BeautifulSoup in bs4 and request in urllib. These two libraries are commonly used in building a web crawler with Python.

```
#import packages
from bs4 import BeautifulSoup
import urllib.request
```

Step 2: Extract the HTML from web page

We need to extract reviews from "https://www.yelp.com/biz/milk-and-cream-cereal-bar-new-york?osq=Ice+Cream". So first, let's save the URL in a variable called url. Then we could access the content on this webpage and save the html in "ourUrl".

```
#the targeted URL
url = 'https://www.yelp.com/biz/milk-and-cream-cereal-bar-new-york?osq=Ice+Cream'
```

```
#use request to open the URL
ourUrl=urllib.request.urlopen(url)
```

Then we use BeautifulSoup to parse the page.

```
#create a BeautifulSoup object, which represents the document as a nested data structure
#parse the page
soup=BeautifulSoup(ourUrl,'html.parser')
```

Now we have the "soup", which is the raw HTML for this website. We could use prettify() to clean the raw data and print it to see the nested structure of HTML in the "soup".

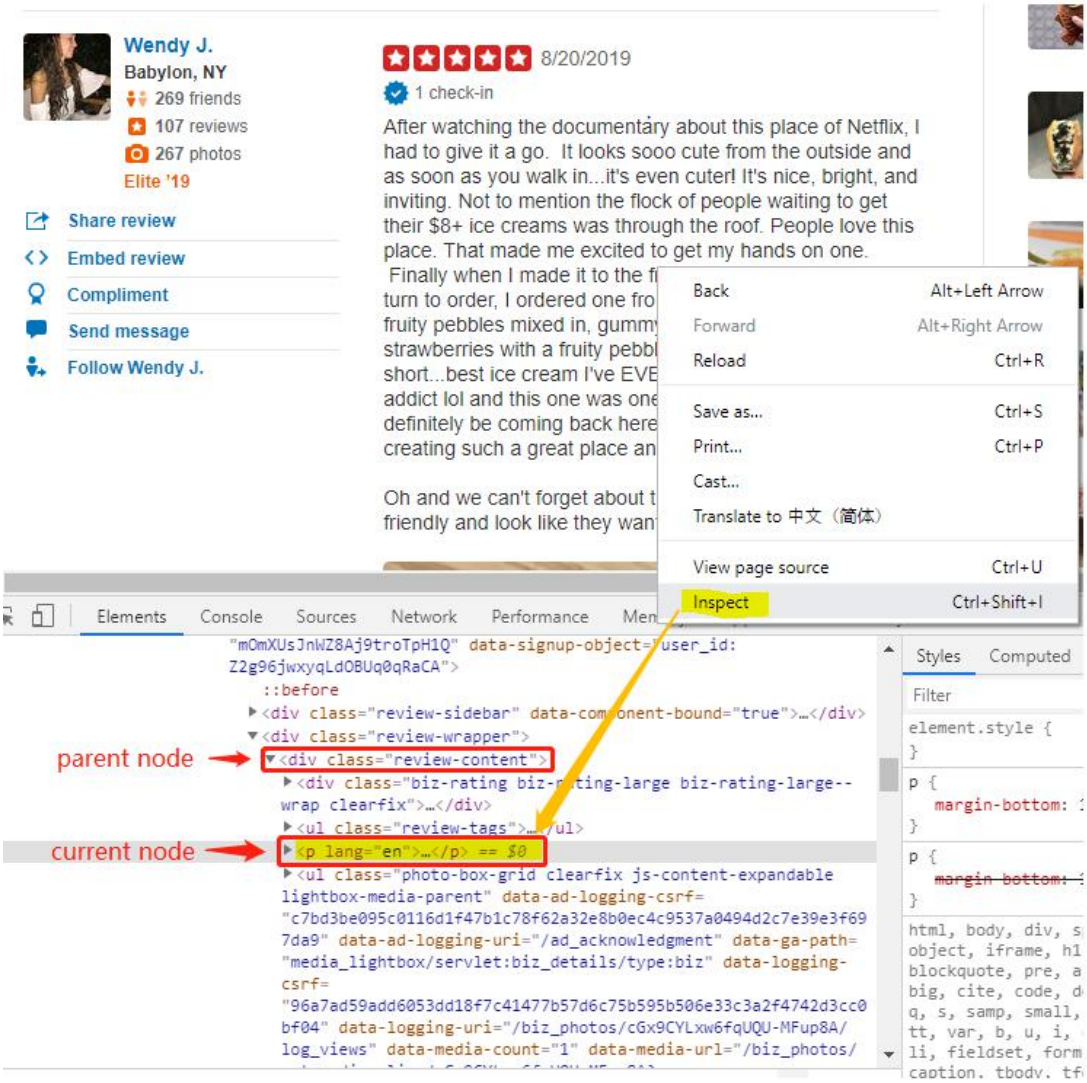
```
# to see what inside the soup
print(soup.prettify())
```

```
<!DOCTYPE HTML>
<!—[if lt IE 7 ]> <html xmlns:fb="http://www.facebook.com/2008/fbml" class="ie6 ie l
tie9 ltie8 no-js" lang="en"> <![endif]—>
<!—[if IE 7 ]>    <html xmlns:fb="http://www.facebook.com/2008/fbml" class="ie7 ie l
tie9 ltie8 no-js" lang="en"> <![endif]—>
<!—[if IE 8 ]>    <html xmlns:fb="http://www.facebook.com/2008/fbml" class="ie8 ie l
tie9 no-js" lang="en"> <![endif]—>
<!—[if IE 9 ]>    <html xmlns:fb="http://www.facebook.com/2008/fbml" class="ie9 ie n
o-js" lang="en"> <![endif]—>
<!—[if (gt IE 9)|!(IE)]><!—>
<html class="no-js" lang="en" xmlns:fb="http://www.facebook.com/2008/fbml">
 <!—<![endif]—>
 <head>
  <script>
   (function() {
                var main = null;

                var main=function(){window.onerror=function(k,a,c,i,f){var j=(documen
t.getElementsByTagName("html")[0].getAttribute("webdriver")=="true"||navigator.userA
```

Step 3: Locate and scrape the reviews

Next, we should find the HTML reviews in this web page, extract them and store them. For each element in the web page, they would have a unique HTML "ID". To check their ID, we would need to INSPECT them in web page.



In this case, the reviews are located under the tag called "p". So we will first use the fuction called find_all() to find the parent node of this reviews. And then locate all elements with the tag "p" under the parent node in a loop. After finding all "p" elements, we would store them in an empty list called "review".

```
review=[]  # create an empty list to store reviews
for i in soup.find_all('div',{'class':'review-content'}):
    per_review=i.find('p')  # extract review
    print(per_review)
    review.append(per_review)  # append review
```

```
<p lang="en">After watching the documentary about this place of Netflix, I had to giv
e it a go.  It looks sooo cute from the outside and as soon as you walk in...it's ev
en cuter! It's nice, bright, and inviting. Not to mention the flock of people waiting
to get their $8+ ice creams was through the roof. People love this place. That made m
e excited to get my hands on one.  Finally when I made it to the from of the line an
d it was my turn to order, I ordered one from the specials menu that has fruity pebbl
es mixed in, gummy bears on top and strawberries with a fruity pebbles cone.  Long s
tory short...best ice cream I've EVER had!  I'm an ice cream addict lol and this one
was one for the books. I will definitely be coming back here!  Props to the owner fo
r creating such a great place and product. <br/><br/>Oh and we can't forget about the
staff. They are all very friendly and look like they want to be there. I love that!</
p>
<p lang="en">fave dessert spot when my friend and i spend the day in the city. the "s
pecials" are delicious and there are very satisfying combinations..my fave being the
chocolate cocoa crunch. the ambience is nice as well, bright lighting and cute visual
s. <br/><br/>just make sure to not sit in the seats right next to the register becaus
e the a/c is way too high and melts the ice cream in your hand all over the place.</p
>
<p lang="en">TLDR: I'm in LOVE with this place! Every flavor of ice cream is so true
```

Now we get all the reviews from that page. Let's see how many reviews have we extracted.

```
len(review)  # how many reviews we collect
```

```
20
```

Step 4: Clean the reviews

You must notice that there still some symbols such as " " at the beginning of each review, "
" in the middle of the reviews and " " at the end of each review. "
" stands for a single line break. We don't need any line break in the reviews so we will need to delete them. Also, " " and " " are the beginning and ending of the HTML and we also need to delete them.

```
#basic clean
New_review=[]  # create an empty list to store new reviews
for each in review:
    new_each=str(each).replace('<br/>','') #remove <br/> with empty, which means delete
    new_each=new_each[13:-4] #remove first 13 strings and last 4 strings, which
                             #are <p lang=' en' > and </p>
    print (new_each)
    New_review.append(new_each)
```

```
After watching the documentary about this place of Netflix, I had to give it a go.   It
looks sooo cute from the outside and as soon as you walk in...it's even cuter! It's nic
e, bright, and inviting. Not to mention the flock of people waiting to get their $8+ ic
e creams was through the roof. People love this place. That made me excited to get my h
ands on one.   Finally when I made it to the from of the line and it was my turn to ord
er, I ordered one from the specials menu that has fruity pebbles mixed in, gummy bears
on top and strawberries with a fruity pebbles cone.   Long story short...best ice cream
I've EVER had!   I'm an ice cream addict lol and this one was one for the books. I will
definitely be coming back here!   Props to the owner for creating such a great place an
d product. Oh and we can't forget about the staff. They are all very friendly and look
like they want to be there. I love that!
fave dessert spot when my friend and i spend the day in the city. the "specials" are de
licious and there are very satisfying combinations..my fave being the chocolate cocoa c
runch. the ambience is nice as well, bright lighting and cute visuals. just make sure t
o not sit in the seats right next to the register because the a/c is way too high and m
elts the ice cream in your hand all over the place.
TLDR: I'm in LOVE with this place! Every flavor of ice cream is so true to taste and I
firmly believe you have to come here at least once for the gram ;) You can't go wrong w
ith any flavor, so I'll be back to sample the rest of the menu.Food: 9/10Each flavor re
```

Now we successfully get all the clean reviews with less than 20 lines of code.

Here is just a demo to scrape 20 reviews from Yelp. But in real case, we may need to face a lot of other situations. For example, we will need steps like pagination to go to other pages and extract the rest reviews for this shop. Or we will need to also scrape down other information like reviewer name, reviewer location, review time, rating, check in.....

To get the above information, we would need to learn more functions and libraries such as selenium or regular expression. It would be interesting to spend more time digging about the challenges in web scraping.

However, if you are looking for some simple ways to do web scraping, Octoparse could be another solution. Octoparse is a powerful web scraping tool which could help you easily obtain information from websites. Check out this tutorial about how to scrape reviews from Yelp with Octoparse. Feel free to contact us when you need a powerful web-scraping tool for your business or project!