



Scraping **Amazon.com** using Python



Challenges:

- Avoiding Captcha checks
- Fetching as much data as needed



Trying to Access **Amazon.com** without Proxy



This site can't be reached

The webpage at [https://www.amazon.com/s?](https://www.amazon.com/s?k=carry+on+luggage&crid=Q5QBWTSPNPFM&sprefix=car%2Caps%2C641&ref=nb_sb_ss_ts-doa-p_2_3)

[k=carry+on+luggage&crid=Q5QBWTSPNPFM&sprefix=car%2Caps%2C641&ref=nb_sb_ss_ts-doa-p_2_3](https://www.amazon.com/s?k=carry+on+luggage&crid=Q5QBWTSPNPFM&sprefix=car%2Caps%2C641&ref=nb_sb_ss_ts-doa-p_2_3) might be temporarily down or it may have moved permanently to a new web address.

ERR_NO_SUPPORTED_PROXIES



Sign up to:
<https://portal.netnut.io/>



Proxies ▾

Scrapers ▾

Datasets ▾

Pricing ▾

Resources ▾

 EN ▾

Register

Login

Set up the Proxy



Pick your preferred Programming Language, Location & URL

The screenshot shows the netnut Proxy Generator interface. At the top right, there is a notification bell, a user profile for 'Zia Ahmad' with a US flag, and a dropdown arrow. Below this is the 'Proxy Generator' section with a terminal icon. It features a 'Python' dropdown menu and a 'New Proxy' button. The configuration fields are: 'Proxy Port' (5959), 'Proxy server' (http://gw.ntnt.io), and 'Proxy user' (blurred). Below these are two location dropdowns: 'Datacenter' (blurred) and 'United States' (with a US flag). A URL input field contains 'https://www.amazon.com/'. At the bottom is a large orange 'Generate' button. Red arrows highlight the 'Python' dropdown, the 'Proxy user' field, the 'United States' dropdown, and the URL input field.

Hey,
Zia Ahmad

Proxy Generator

Python

New Proxy

Proxy Port
5959

Proxy server
http://gw.ntnt.io

Proxy user

Datacenter

United States

https://www.amazon.com/

Generate



- **Import Libraries**
- **Paste the Data that we got from NetNut.io**

```
from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
from bs4 import BeautifulSoup
import time
```

```
username = 'zia@ntnt.io'
password = 'r8R@M'
server = 'gw.ntnt.io'
port = '5959'
proxy_url = f'http://{username}:{password}@{server}:{port}'
```



- **Setting up WebDrivers**
- **Hitting the URL of [Amazon.com](https://www.amazon.com/) to get the data**

```
capabilities = webdriver.DesiredCapabilities.CHROME
options = webdriver.ChromeOptions()
options.add_argument(f'--proxy-server={proxy_url}')
driver = webdriver.Chrome(ChromeDriverManager().install(),
    chrome_options = options, desired_capabilities = capabilities)

keyword = "Cars"
url = f'https://www.amazon.com/'
driver.get(url)
time.sleep(3)
driver.find_element_by_xpath('//*[@id="b"]').click()
time.sleep(3)
driver.find_element_by_id("twotabsearchtextbox").send_keys(keyword)
driver.find_element_by_id('nav-search-submit-button').click()
```



- Parsing the required Data
- Printing the
 - Product name
 - Price

```
soup = BeautifulSoup(driver.page_source, 'html.parser')
post_cards = soup.find_all("div", "a-section a-spacing-small a-spacing-top-small")
data = []
for i in post_cards:
    dct = {}
    try:
        dct['Price'] = i.find(class_="a-offscreen").get_text()
        dct['Title'] = i.find(class_="a-size-medium a-color-base a-text-normal").get_text()
        data.append(dct)
    except:
        print("Didn't find the Text")

pd.DataFrame(data)
```



Sample Output

Price		Title
\$3.79		Cars
\$13.99	Hot Wheels Set of 10 Toy Cars & Trucks in 1:64...	
\$21.97	Mattel Disney and Pixar Cars Transforming Mack...	
\$23.00	Mattel Disney and Pixar Cars Mini Racers Set o...	
\$23.00	Matchbox Cars, 20-Pack of 1:64 Scale Die-Cast ...	
\$13.49	Mattel Disney and Pixar Cars Mini Racers 3-Pac...	
\$22.15	Mattel Disney and Pixar Cars Set of 5 Collecti...	
\$38.87	Carrera First Disney/Pixar Cars - Slot Car Rac...	
\$40.40	Mattel Disney and Pixar Cars Set of 10 Die-Cas...	
\$3.79		Cars 3 (Theatrical)
\$3.79		Cars 2



Scraping Amazon.com using Python

Hit Follow!



Save for later!



netnut.io