

# Web Scrapping using Selenium

**Code by : KRISH GOYAL**

**Email\_id : goyal.krish0522@gmail.com**

In [2]: `# pip install selenium`

```
Collecting selenium
  Using cached selenium-4.15.2-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: urllib3<3,>=1.26 in c:\users\goyal\anaconda3\lib\site-packages
(from urllib3[socks]<3,>=1.26->selenium) (1.26.16)
Collecting trio~=0.17 (from selenium)
  Downloading trio-0.23.1-py3-none-any.whl.metadata (4.9 kB)
Collecting trio-websocket~=0.9 (from selenium)
  Downloading trio_websocket-0.11.1-py3-none-any.whl.metadata (4.7 kB)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\goyal\anaconda3\lib\site-packag
es (from selenium) (2023.7.22)
Requirement already satisfied: attrs>=20.1.0 in c:\users\goyal\anaconda3\lib\site-packages (f
rom trio~=0.17->selenium) (23.1.0)
Requirement already satisfied: sortedcontainers in c:\users\goyal\anaconda3\lib\site-packages
(from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: idna in c:\users\goyal\anaconda3\lib\site-packages (from trio~
=0.17->selenium) (3.4)
Collecting outcome (from trio~=0.17->selenium)
  Downloading outcome-1.3.0.post0-py2.py3-none-any.whl.metadata (2.6 kB)
Collecting sniffio>=1.3.0 (from trio~=0.17->selenium)
  Downloading sniffio-1.3.0-py3-none-any.whl (10 kB)
Requirement already satisfied: cffi>=1.14 in c:\users\goyal\anaconda3\lib\site-packages (from
trio~=0.17->selenium) (1.15.1)
Collecting wsproto>=0.14 (from trio-websocket~=0.9->selenium)
  Downloading wsproto-1.2.0-py3-none-any.whl (24 kB)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\users\goyal\anaconda3\lib\si
te-packages (from urllib3[socks]<3,>=1.26->selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\users\goyal\anaconda3\lib\site-packages (from
cffi>=1.14->trio~=0.17->selenium) (2.21)
Collecting h11<1,>=0.9.0 (from wsproto>=0.14->trio-websocket~=0.9->selenium)
  Using cached h11-0.14.0-py3-none-any.whl (58 kB)
Using cached selenium-4.15.2-py3-none-any.whl (10.2 MB)
Downloading trio-0.23.1-py3-none-any.whl (448 kB)
----- 0.0/448.3 kB ? eta -:-:--
----- 10.2/448.3 kB ? eta -:-:--
----- 61.4/448.3 kB 656.4 kB/s eta 0:00:01
----- 81.9/448.3 kB 657.6 kB/s eta 0:00:01
----- 163.8/448.3 kB 821.4 kB/s eta 0:00:01
----- 194.6/448.3 kB 841.6 kB/s eta 0:00:01
----- 286.7/448.3 kB 983.9 kB/s eta 0:00:01
----- 337.9/448.3 kB 1.1 MB/s eta 0:00:01
----- 337.9/448.3 kB 1.1 MB/s eta 0:00:01
----- 448.3/448.3 kB 1.1 MB/s eta 0:00:00
Downloading trio_websocket-0.11.1-py3-none-any.whl (17 kB)
Downloading outcome-1.3.0.post0-py2.py3-none-any.whl (10 kB)
Installing collected packages: sniffio, outcome, h11, wsproto, trio, trio-websocket, selenium
  Attempting uninstall: sniffio
    Found existing installation: sniffio 1.2.0
    Uninstalling sniffio-1.2.0:
      Successfully uninstalled sniffio-1.2.0
Successfully installed h11-0.14.0 outcome-1.3.0.post0 selenium-4.15.2 sniffio-1.3.0 trio-0.2
3.1 trio-websocket-0.11.1 wsproto-1.2.0
Note: you may need to restart the kernel to use updated packages.
```

In [42]: `from selenium import webdriver`  
`from selenium.webdriver.common.by import By`

```

from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from bs4 import BeautifulSoup

# Set your username and password
username = "lavender1032"
password = "lavender1032"

# Replace 'your_url' with the actual URL of the webpage you want to scrape after logging in
login_url = 'https://windscribe.com/login'
target_url = 'https://windscribe.com/features'

# Create a new instance of the Chrome driver
driver = webdriver.Chrome()

# Navigate to the Login page
driver.get(login_url)

# Find and fill in the username and password fields
username_field = driver.find_element(By.ID, "username")
password_field = driver.find_element(By.ID, "pass")
username_field.send_keys(username)
password_field.send_keys(password)

# Submit the Login form
login_button = driver.find_element(By.ID, "login_button")
login_button.click()

# Wait for the URL to change after Login
WebDriverWait(driver, 20).until(EC.url_changes)

# Print current URL for debugging
print("Current URL:", driver.current_url)

# Now you are logged in, navigate to the target page
driver.get(target_url)

# Wait for the content to load if necessary (add WebDriverWait if needed)

# Now you can proceed to scrape items under Header 2
soup = BeautifulSoup(driver.page_source, 'html.parser')

# Create a dictionary to store the scraped data
scraped_data = {}

# Example: Print text of all elements under Header 2
header_2_elements = soup.select('h2 ~ *')
for element in header_2_elements:
    header_text = element.find_previous('h2').text.strip()
    element_text = element.text.strip()
    print(element_text)
    scraped_data.setdefault(header_text, []).append(element_text)

# Close the browser
driver.quit()

```

Current URL: <https://windscribe.com/myaccount>  
Large Network  
Servers in over 63 countries and 110 cities  
R.O.B.E.R.T.  
Blocks IPs and domains (ads) of your choice on all devices  
No Identifying Logs  
We can't personally identify you based on IP and timestamp  
Strongest Encryption  
We use AES-256 cipher with SHA512 auth and a 4096-bit RSA key  
Config Generators  
Generate OpenVPN, IKEv2, and WireGuard® configs for all your devices  
ScribeForce  
Team Accounts with a cool name  
Simple Clients  
Minimalist clients for all platforms that will make getting started a breeze  
Use for Free  
Generous free plans are available for those who are on a tight budget  
Static IPs  
Some IPs are not meant to change, they just stay the same  
Port Forwarding  
Access internal resources securely  
Split Tunneling  
Choose which apps go over the VPN, and which don't  
Desktop App Features  
Windows  
Mac  
Firewall  
Blocks all connectivity outside the tunnel to eliminate all types of leaks  
Secure Hotspot  
Turn your computer into a secure Wifi router  
Proxy Gateway  
Create a proxy server on your network for other devices  
Flexible Connectivity  
Connect via IKEv2, OpenVPN UDP, TCP or Stealth on a wide range of ports  
Browser Extension Features  
Auto Pilot  
Automatically picks the best location for you  
Double Hop  
Proxy your connection through any 2 servers on the Windscribe network  
Ad Blocking  
Blocks ads and trackers that follow you across the web  
Time Warp  
Changes your timezone so appear to be in the country you're connected to  
Cookie Monster  
Keeps track of cookies that are set, and deletes them on tab close  
Split Personality  
Randomly rotates your user agent to reduce the chance of basic fingerprinting  
WebRTC Slayer  
Blocks WebRTC to prevent leaks outside of the tunnel  
Location Warp  
Spoofs your GPS to the location of the connected proxy

In [43]: `scraped_data`

```
Out[43]: {'General Features': ['Large Network\nServers in over 63 countries and 110 cities',
    'R.O.B.E.R.T.\nBlocks IPs and domains (ads) of your choice on all devices',
    'No Identifying Logs\nWe can't personally identify you based on IP and timestamp',
    'Strongest Encryption\nWe use AES-256 cipher with SHA512 auth and a 4096-bit RSA key',
    'Config Generators\nGenerate OpenVPN, IKEv2, and WireGuard® configs for all your devices',
    'ScribeForce\nTeam Accounts with a cool name',
    'Simple Clients\nMinimalist clients for all platforms that will make getting started a breeze',
    'Use for Free\nGenerous free plans are available for those who are on a tight budget',
    'Static IPs\nSome IPs are not meant to change, they just stay the same',
    'Port Forwarding\nAccess internal resources securely',
    'Split Tunneling\nChoose which apps go over the VPN, and which don't',
    'Desktop App Features'],
    'Desktop App Features': ['Windows',
    'Mac',
    'Firewall\nBlocks all connectivity outside the tunnel to eliminate all types of leaks',
    'Secure Hotspot\nTurn your computer into a secure Wifi router',
    'Proxy Gateway\nCreate a proxy server on your network for other devices',
    'Flexible Connectivity\nConnect via IKEv2, OpenVPN UDP, TCP or Stealth on a wide range of ports',
    'Browser Extension Features'],
    'Browser Extension Features': ['Auto Pilot\nAutomatically picks the best location for you',
    'Double Hop\nProxy your connection through any 2 servers on the Windscribe network',
    'Ad Blocking\nBlocks ads and trackers that follow you across the web',
    'Time Warp\nChanges your timezone so appear to be in the country you're connected to',
    'Cookie Monster\nKeeps track of cookies that are set, and deletes them on tab close',
    'Split Personality\nRandomly rotates your user agent to reduce the chance of basic fingerprinting',
    'WebRTC Slayer\nBlocks WebRTC to prevent leaks outside of the tunnel',
    'Location Warp\nSpoofs your GPS to the location of the connected proxy']}]
```

```
In [44]: import pandas as pd

# Assuming scraped_data is the dictionary you want to convert to a DataFrame
df = pd.DataFrame.from_dict(scraped_data, orient='index').transpose()
df = df[['General Features']]
df = df.iloc[:-1]
# Print the DataFrame
df
```

Out[44]:

	General Features
0	Large Network\nServers in over 63 countries an...
1	R.O.B.E.R.T.\nBlocks IPs and domains (ads) of ...
2	No Identifying Logs\nWe can't personally ident...
3	Strongest Encryption\nWe use AES-256 cipher wi...
4	Config Generators\nGenerate OpenVPN, IKEv2, an...
5	ScribeForce\nTeam Accounts with a cool name
6	Simple Clients\nMinimalist clients for all pla...
7	Use for Free\nGenerous free plans are availabl...
8	Static IPs\nSome IPs are not meant to change, ...
9	Port Forwarding\nAccess internal resources sec...
10	Split Tunneling\nChoose which apps go over the...

```
In [45]: df['Feature Names'] = df['General Features'].apply(lambda x: x.split('\n')[0])
df = df[['Feature Names']]
df
```

Out[45]:

Feature Names	
0	Large Network
1	R.O.B.E.R.T.
2	No Identifying Logs
3	Strongest Encryption
4	Config Generators
5	ScribeForce
6	Simple Clients
7	Use for Free
8	Static IPs
9	Port Forwarding
10	Split Tunneling

- This is the data extracted after login in the website <https://windscribe.com> from the General Features --> Feature Names
- **Username :** "lavender1032"
- **Password :** "lavender1032"