ScrapeHero a data company

How To Rotate Proxies and IP Addresses using Python 3

A common problem faced by web scrapers is getting blocked by websites while scraping them. There are many techniques to prevent getting blocked, like

- Rotating IP addresses
- Using Proxies
- Rotating and Spoofing user agents
- Using headless browsers
- Reducing the crawling rate

etc.

> Learn More: How to prevent getting blacklisted while scraping

Using proxies and rotating IP addresses in combination with rotating user agents can help you get scrapers past most of the anti-scraping measures and prevent being detected as a scraper.

the website that you are not a single 'bot' or a person accessing the website, but multiple 'real' users accessing the website from multiple locations. If you do it right, the chances of getting blocked are minimal.

In this blog post we will show you how to send your requests to a website using a proxy, and then we'll show you how to send these requests through multiple IP addresses or proxies.

How to send requests through a Proxy in Python 3 using Requests

If you are using Python-Requests, you can send requests through a proxy by configuring the `proxies` argument. For example

```
import requests

proxies = {
  'http': 'http://10.10.1.10:3128',
  'https': 'http://10.10.1.10:1080'
}

requests.get('http://example.org',
```

Let's find a proxy

There are many websites dedicated to providing free proxies on the internet. One such site is https://free-proxy-list.net/. Let's go there and pick a proxy that supports https (as we are going to test this on an https website).

Here is our proxy –

IP: 207.148.1.212 Port: 8080

Note:

This proxy might not work when you test it. You should pick another proxy from the website if it doesn't work.

Now let's make a request to HTTPBin's IP endpoint and test if the request went through the proxy

```
import requests
url = 'https://httpbin.org/ip'
proxies = {
    "http": 'http://209.50.52.162:9
    "https": 'http://209.50.52.162:
}
response = requests.get(url,proxies
print(response.json())
```

```
{'origin': '209.50.52.162'}
```

You can see that the request went through the proxy. Let's get to sending requests through a pool of IP addresses.

Rotating Requests through a pool of Proxies in Python 3

We'll gather a list of some active proxies from https://free-proxy-list.net/. You can also use private proxies if you have access to them.

You can make this list by manually copy and pasting, or automate this by using a scraper (If you don't want to go through the hassle of copy and pasting every time the proxies you have gets removed). You can write a script to grab all the proxies you need and construct this list dynamically every time you initialize your web scraper. Once you have the list of Proxy IPs to rotate, the rest is easy.

We have written some code to pick up IPs automatically by scraping. (This code could change when the website updates its structure)

```
from lxml.html import fromstring
def get_proxies():
    url = 'https://free-proxy-list.
    response = requests.get(url)
    parser = fromstring(response.te
    proxies = set()
    for i in parser.xpath('//tbody/
        if i.xpath('.//td[7][contai
            #Grabbing IP and corres
            proxy = ":".join([i.xpa
            proxies.add(proxy)
    return proxies
```

The function `get_proxies` will return a set of proxy strings that can be passed to the request object as proxy config.

```
proxies = get_proxies()
print(proxies)
```

```
{'121.129.127.209:80', '124.41.215.
```

Now that we have the list of Proxy IP Addresses in a variable `proxies`, we'll go ahead and rotate it using a Round Robin method.

```
import requests
from itertools import cycle
import traceback
```

```
proxies = get_proxies()
proxy_pool = cycle(proxies)

url = 'https://httpbin.org/ip'
for i in range(1,11):
    #Get a proxy from the pool
    proxy = next(proxy_pool)
    print("Request #%d"%i)
    try:
        response = requests.get(url
        print(response.json())
    except:
        #Most free proxies will oft
        #We will just skip retries
        print("Skipping. Connnectio
```

```
Request #1
{'origin': '121.129.127.209'}
Request #2
{'origin': '124.41.215.238'}
Request #3
{'origin': '185.93.3.123'}
Request #4
{'origin': '194.182.64.67'}
Request #5
Skipping. Connnection error
Request #6
{'origin': '163.172.175.210'}
Request #7
{'origin': '13.92.196.150'}
Request #8
{'origin': '121.129.127.209'}
Request #9
{'origin': '124.41.215.238'}
```

Okay – it worked. Request #5 had a connection error probably because the free proxy we grabbed was overloaded with users trying to get their proxy traffic through. Below is the full code to do this.

Full Code

```
from lxml.html import fromstring
import requests
from itertools import cycle
import traceback

def get_proxies():
    url = 'https://free-proxy-list.
    response = requests.get(url)
    parser = fromstring(response.te
    proxies = set()
    for i in parser.xpath('//tbody/
        if i.xpath('.//td[7][contai
            proxy = ":".join([i.xpa
            proxies.add(proxy)
    return proxies


#If you are copy pasting proxy ips,
#proxies = ['121.129.127.209:80', '
proxies = get_proxies()
proxy_pool = cycle(proxies)

url = 'https://httpbin.org/ip'
for i in range(1,11):
```

```
    print("Request #%d"%i)
    try:
        response = requests.get(url
        print(response.json())
    except:
        #Most free proxies will oft
        #We will just skip retries
        print("Skipping. Connnectio
```

Rotating Proxies in Scrapy

Scrapy does not have built in proxy rotation. There are many middlewares in scrapy for rotating proxies or ip address in scrapy. We have found scrapy-rotating-proxies to be the most useful among them.

Install scrapy-rotating-proxies using

```
pip install scrapy-rotating-proxies
```

In your scrapy project's settings.py add,

```
DOWNLOADER_MIDDLEWARES = {
    'rotating_proxies.middlewares.R
}

ROTATING_PROXY_LIST = [
    'proxy1.com:8000',
    'proxy2.com:8031',
```

As an alternative to ROTATING_PROXY_LIST, you can specify a ROTATING_PROXY_LIST_PATH options with a path to a file with proxies, one per line: ROTATING_PROXY_LIST_PATH = '/my/path/proxies.txt'

You can read more about this middleware on its github repo.

5 Things to keep in mind while using proxies and rotating IP addresses

Here are a couple tips that you should remember:

1. Avoid Using Proxy IP addresses that are in a sequence

Even the simplest anti-scraping plugins can detect that you are a scraper if the requests come from IP addresses that are continuous or belong to the same range like this:

64.233.160.0

64.233.160.2

64.233.160.3

Some websites have gone as far as blocking the entire providers like AWS and have even blocked entire countries.

2. If you are using free proxies – automate

Free proxies tend to die out soon, mostly in days or hours and would expire before the scraping even completes. To prevent that from disrupting your scrapers, write some code that would automatically pick up and refresh the proxy list you use for scraping with working IP addresses. This will save you a lot of time and frustration.

3. Use Elite Proxies whenever possible if you are using Free Proxies (or even if you are paying for proxies)

All proxies aren't the same. There are mainly three types of proxies available in the internet.

computer and the internet and redirects your requests and responses without modifying them. It sends your real IP address in the HTTP_X_FORWARDED_FOR header, this means a website that does not only determine your REMOTE_ADDR but also checks for specific proxy headers that will still know your real IP address. The HTTP_VIA header is also sent, revealing that you are using a proxy server.

2. **Anonymous Proxy** – An anonymous proxy does not send your real IP address in the HTTP_X_FORWARDED_FOR header, instead, it submits the IP address of the proxy or it'll just be blank. The HTTP_VIA header is sent with a transparent proxy, which would reveal you are using a proxy server. An anonymous proxy server does not tell websites your real IP address anymore. This can be helpful to just keep your privacy on the internet. The website can still see you are using a proxy server, but in the end, it does not really matter as long as the proxy server does not disclose your real IP address.

will be detected and blocked.

3. **Elite Proxy** – An elite proxy only sends REMOTE_ADDR header while the other headers are empty. It will make you seem like a regular internet user who is not using a proxy at all. An elite proxy server is ideal to pass any restrictions on the internet and to protect your privacy to the fullest extent. You will seem like a regular internet user who lives in the country that your proxy server is running in.

Elite Proxies are your best option as they are hard to be detected. Use anonymous proxies if it's just to keep your privacy on the internet. Lastly, use transparent proxies – although the chances of success are very low.

4. Get Premium Proxies if you are Scraping Thousands of Pages

Free proxies available on the internet are always abused and end up being in blacklists used by anti-scraping tools and web servers. If you are doing serious large-scale data extraction, you should pay for

you.

5. Use IP Rotation in combination with Rotating User Agents

IP rotation on its own can help you get past some anti-scraping measures. If you find yourself being banned even after using rotating proxies, a good solution is adding header spoofing and rotation.

That's all we've got to say. Happy Scraping 🙂

Continue Reading ..

How to Parse Unstructured Addresses using Python and Google GeoCoding API

If you have come across a large number of freeform address as a single string, for example - "9 Downing St Westminster London SW1A, UK", you know how hard it would be to validate, compare...