

# Scrapy with Privoxy and Tor: how to renew IP

Asked 7 years, 8 months ago

Modified 3 years, 11 months ago

Viewed 20k times

▲

22

▼

🔖

🕒

I am dealing with Scrapy, Privoxy and Tor. I have all installed and properly working. But Tor connects with the same IP everytime, so I can easily be banned. Is it possible to tell Tor to reconnect each X seconds or connections?

Thanks!

EDIT about the configuration: For the user agent pool i did this: <http://tangww.com/2013/06/UsingRandomAgent/> (I had to put a `_init_.py` file as it is said in the comments), and for the Privoxy and Tor I followed <http://www.andrewwatters.com/privoxy/> (I had to create the private user and private group manually with the terminal). It worked :)

My spider is this:

```
from scrapy.contrib.spiders import CrawlSpider
from scrapy.selector import Selector
from scrapy.http import Request

class YourCrawler(CrawlSpider):
    name = "spider_name"
    start_urls = [
        'https://example.com/listviews/titles.php',
    ]
    allowed_domains = ["example.com"]

    def parse(self, response):
        # go to the urls in the list
        s = Selector(response)
        page_list_urls = s.xpath('///*[
[id="tab7"]/article/header/h2/a/@href').extract()
        for url in page_list_urls:
            yield Request(response.urljoin(url),
callback=self.parse_following_urls, dont_filter=True)

        # Return back and go to bext page in div#paginat ul li.next
a::attr(href) and begin again
        next_page = response.css('ul.pagin li.presente ~ li
a::attr(href)').extract_first()
        if next_page is not None:
            next_page = response.urljoin(next_page)
            yield Request(next_page, callback=self.parse)

        # For the urls in the list, go inside, and in div#main, take the div.ficha
> div.caracteristicas > ul > li
        def parse_following_urls(self, response):
            #Parsing rules go here
            for each_book in response.css('main#main'):
                yield {
                    'editor': each_book.css('header.datos1 > ul > li > h5 >
a::text').extract(),
                }
```

In settings.py I have an user agent rotation and privoxy:

```
DOWNLOADER_MIDDLEWARES = {
    #user agent
    'scrapy.contrib.downloadermiddleware.useragent.UserAgentMiddleware' :
None,
    'spider_name.comm.rotate_useragent.RotateUserAgentMiddleware' :400,
    #privoxy
    'scrapy.downloadermiddlewares.httpproxy.HttpProxyMiddleware': 110,
    'spider_name.middlewares.ProxyMiddleware': 100
}
```

In middlewares.py I added:

```
class ProxyMiddleware(object):
    def process_request(self, request, spider):
        request.meta['proxy'] = 'http://127.0.0.1:8118'
        spider.log('Proxy : %s' % request.meta['proxy'])
```

And I think that's all...

EDIT II ---

Ok, I changed my middlewares.py file as in the blog @Tomáš Linhart said from:

```
class ProxyMiddleware(object):
    def process_request(self, request, spider):
        request.meta['proxy'] = 'http://127.0.0.1:8118'
        spider.log('Proxy : %s' % request.meta['proxy'])
```

To

```
from stem import Signal
from stem.control import Controller

class ProxyMiddleware(object):
    def process_request(self, request, spider):
        request.meta['proxy'] = 'http://127.0.0.1:8118'
        spider.log('Proxy : %s' % request.meta['proxy'])

    def set_new_ip():
        with Controller.from_port(port=9051) as controller:
            controller.authenticate(password='tor_password')
            controller.signal(Signal.NEWNYM)
```

But now is really slow, and doesn't appear to change the ip... I did it ok or is something wrong?

python

web-scraping

scrapy

tor

Share

Improve this question

Follow

edited Jul 10, 2017 at 16:04

asked Jul 10, 2017 at 10:41

user7499416

this project on github explains how to scrap anonymously [github.com/WiliTest/...](#) – J. Does Dec 21, 2017 at 17:21

2 Answers

Sorted by:

Highest score (default)

▲

12

▼

🔖

✓

🕒

This [blog post](#) might help you a bit as it deals with the same issue.

**EDIT:** Based on concrete requirement (new IP for each request or after *N* requests), put appropriate call to `set_new_ip` in `process_request` method of the middleware. Note, however, that call to `set_new_ip` function doesn't have to always ensure new IP (there's a link to the FAQ with explanation).

**EDIT2:** The module with `ProxyMiddleware` class would look like this:

```
from stem import Signal
from stem.control import Controller

def _set_new_ip():
    with Controller.from_port(port=9051) as controller:
        controller.authenticate(password='tor_password')
        controller.signal(Signal.NEWNYM)

class ProxyMiddleware(object):
    def process_request(self, request, spider):
        _set_new_ip()
        request.meta['proxy'] = 'http://127.0.0.1:8118'
        spider.log('Proxy : %s' % request.meta['proxy'])
```


Share

Improve this answer

Follow

edited Jul 11, 2017 at 9:51

answered Jul 10, 2017 at 10:51



Tomáš Linhart

10.2k13041

Hm. So I have to install stem, and then, as in **Changing IP address** use the `def set_new_ip():` function inside my spider code importing stem. Is this correct? Or the `def set_new_ip():` has to go in a middleware? – user7499416 Jul 10, 2017 at 11:00

@Nikita What exactly is your requirement? To get new IP for each spider run, or even for each request? – Tomáš Linhart Jul 10, 2017 at 11:15

For each request or every x number of request. Is to make bore difficult to be banned by IP. – user7499416 Jul 10, 2017 at 11:21

When you say «spider run», you mean each run of the loop inside the spider, or each time the spider is opened? – user7499416 Jul 10, 2017 at 11:39

And how do you have the Scrapy + Privoxy + Tor combination set up? – Tomáš Linhart Jul 10, 2017 at 11:39

▲

12

▼

🔖

🕒

But Tor connects with the same IP everytime

That is a [documented Tor feature](#):

An important thing to note is that **a new circuit does not necessarily mean a new IP address**. Paths are randomly selected based on heuristics like speed and stability. There are only so many large exits in the Tor network, so it's not uncommon to reuse an exit you have had previously.

That's the reason why using the code below can result in reusing the same IP address again.

```
from stem import Signal
from stem.control import Controller

with Controller.from_port(port=9051) as controller:
    controller.authenticate(password='tor_password')
    controller.signal(Signal.NEWNYM)
```

<https://github.com/DusanMadar/TorIpChanger> helps you to manage this behavior. Disclaimer - I wrote TorIpChanger .

I've also put together a guide on how to use Python with Tor and Privoxy:  
<https://gist.github.com/DusanMadar/8d11026b7ce0bce6a67f7dd87b999f6b>.

Here's an example of how you can use `TorIpChanger` (`pip install toripchanger`) in your `ProxyMiddleware`.

```
from toripchanger import TorIpChanger

# A Tor IP will be reused only after 10 different IPs were used.
ip_changer = TorIpChanger(reuse_threshold=10)

class ProxyMiddleware(object):
    def process_request(self, request, spider):
        ip_changer.get_new_ip()
        request.meta['proxy'] = 'http://127.0.0.1:8118'
        spider.log('Proxy : %s' % request.meta['proxy'])
```

Or, if you want to use a different IP after 10 requests, you can do something like below.

```
from toripchanger import TorIpChanger

# A Tor IP will be reused only after 10 different IPs were used.
ip_changer = TorIpChanger(reuse_threshold=10)

class ProxyMiddleware(object):
    _requests_count = 0

    def process_request(self, request, spider):
        self._requests_count += 1
        if self._requests_count > 10:
            self._requests_count = 0
            ip_changer.get_new_ip()

        request.meta['proxy'] = 'http://127.0.0.1:8118'
        spider.log('Proxy : %s' % request.meta['proxy'])
```

Share Improve this answer Follow

edited Apr 20, 2021 at 21:42

answered Mar 7, 2018 at 10:37

 **Dušan Maďar**

9,92965670