

How to use proxy for specific url in Scrapy spider?

Asked 7 years, 2 months ago Modified 6 years, 10 months ago Viewed 4k times

I want to use proxy for only few specific domain. I check [this](#), [this](#) and [this](#). If I understand correctly setting proxy using middleware will set the proxy for all requests.

1 How can I set proxy for specific url before the spider request is sent?

Currently my spider is working fine with following implementation:



CoreSpider.py

```
class CoreSpider(scrapy.Spider):
    name = "final"
    def __init__(self):
        self.start_urls = self.read_url()
        self.rules = (
            Rule(
                LinkExtractor(
                    unique=True,
                ),
                callback='parse',
                follow=True
            ),
        )

    def read_url(self):
        urlList = []
        for filename in glob.glob(os.path.join("/root/Public/company_profiler/seed_list", '*.list')):
            with open(filename, "r") as f:
                for line in f.readlines():
                    url = re.sub('\n', '', line)
                    if "http" not in url:
                        url = "http://" + url
                    # print(url)
                    urlList.append(url)

        return urlList

    def parse(self, response):
        print("URL is: ", response.url)
        print("User agent is : ", response.request.headers['User-Agent'])
        filename = '/root/Public/company_profiler/crawled_page/%s.html' % response.url
        article = Extractor(extractor='LargestContentExtractor',
            html=response.body).getText()
        print("Article is :", article)
        if len(article.split("\n")) < 5:
            print("Skipping to next url : ", article.split("\n"))
        else:
            print("Continue parsing: ", article.split("\n"))
            ContentHandler_copy.ContentHandler_copy.start(article,
                response.url)
```

and settings.py

```
DOWNLOADER_MIDDLEWARES = {
    'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware': None,
    'random_useragent.RandomUserAgentMiddleware': 320
}
```

I am running spider by calling it via script RunSpider.py

RunSpider.py

```
from CoreSpider import CoreSpider
from scrapy.crawler import CrawlerProcess
from scrapy.utils.project import get_project_settings

process = CrawlerProcess(get_project_settings())
process.crawl(CoreSpider)
process.start()
```

Update: CoreSpider.py

```
class CoreSpider(scrapy.Spider):
    name = "final"
    def __init__(self):
        self.start_urls = self.read_url()
        self.rules = (
            Rule(LinkExtractor(unique=True), callback='parse', follow=True,
                process_request='process_request'),
```

```
)

def process_request(self, request, spider):
    print("Request is :", request) ### Not printing anything
    if 'xxx' in request.url: # <-- set proxy for this URL?
        meta = request.get('meta', {})
        meta.update({'proxy': 'https://159.8.18.178:8080'})
        return request.replace(meta=meta)
    return request
    .....
```



I also tried setting proxy like this in process_request method, but failed.

```
request.meta['proxy'] = "https://159.8.18.178:8080"
```

Thanks in advance.



python python-3.x scrapy

Share Improve this question Follow

edited Jan 8, 2018 at 10:52

asked Jan 8, 2018 at 8:58



Om Prakash 2,901 4 31 50

3 Answers



Sorted by: Highest score (default)



1

To use proxy per request, specify proxy attribute of Request 's meta as per [documentation](#). In case of CrawlSpider , you'll want to supply process_request argument to the Rule . In that method, apply the above (i.e. setting meta['proxy']) selectively based on the request URL and return modified request with meta filled.



EDIT: Replace the rule definition



```
self.rules = (
    Rule(LinkExtractor(unique=True), callback='parse', follow=True),
)
```



with

```
self.rules = (
    Rule(LinkExtractor(unique=True), callback='parse', follow=True,
    process_request='process_request'),
)
```



and define new method process_request in your CoreSpider class:

```
def process_request(self, request):
    if 'xxx' in request.url: # <-- set proxy for this URL?
        meta = request.get('meta', {})
        meta.update({'proxy': 'your_proxy'})
        return request.replace(meta=meta)
    return request
```



EDIT2: I think the problem might be caused by having start_urls and rules definition buried in the constructor:



```
...
def __init__(self):
    self.start_urls = self.read_url()
    self.rules = (
        Rule(LinkExtractor(unique=True), callback='parse', follow=True,
        process_request='process_request'),
    )
...
```

The correct way is to have these attributes as class attributes, i.e.

```
class CoreSpider(scrapy.Spider):
    name = "final"
    start_urls = self.read_url()
    rules = (
        Rule(LinkExtractor(unique=True), callback='parse', follow=True,
        process_request='process_request'),
    )
```

As for start_urls , in cases where you need something more complicated (e.g. reading URLs from external file), it might be better and more readable to define [start_requests](#) to yield the Request s.



I didn't get you. How do I supply `process_request` argument to Rule? Could you please give example? I tried `class ProxyMiddleware(object):` `def process_request(self, request, spider):` `print("Here in proxy middleware.")` `request.meta['proxy'] = "proxy:port"` in `middlewares.py` . But it is not working. – [Om Prakash](#) Jan 8, 2018 at 9:29

It is not working. I tried printing the request in `process_request` method before checking condition, but it is not even printing anything. Perhaps `process_request` is not getting called for each url. – [Om Prakash](#) Jan 8, 2018 at 10:39

Could you edit the question and post update source code so I can check? – [Tomáš Linhart](#) Jan 8, 2018 at 10:42

My bad! `process_request` was overridden for dynamic user-agent for every request. I made the suggested changes there and boom! now it is working. – [Om Prakash](#) Feb 5, 2018 at 9:57



a standalone method. No middleware.

1



```
urls = [url, url, ..., url]
class TestSpider(scrapy.Spider):
    name = 'test'
    allowed_domains = ['test.com']
    # start_urls = urls # invalid, override by start_requests

    def start_requests(self):
        for url in urls:
            # handle each individual url with or without proxy
            # if url in ['no1.com', 'no2.com', 'no3.com']:
            if url == 'www.no_proxy.com':
                meta_proxy = '' # do not use proxy for this url
            else:
                meta_proxy = "http://127.0.0.1:8888"
            yield scrapy.Request(url=url, callback=self.parse, meta={'proxy':
meta_proxy})

    def parse(self, response):
        title = response.xpath('..//title/text()').extract_first()
        yield {'title': title}
```



usage:

```
$scrapy runspider test.py -o test.json -s CONCURRENT_REQUESTS_PER_DOMAIN=100 -s
CONCURRENT_REQUESTS=100
```

Disclaimer:

I don't know if it will slow down crawling speed or not, since it is iterating urls one by one. I do not have tons of testing sites for the moment. Hope someone who use this code will leave a comment to see what they got.



I give you an example for using proxy with specific url

0



```
link = 'https://www.example.com/'
request = Request(link, callback=self.parse_url)
request.meta['proxy'] = "http://PROXYIP:PROXYPORT"
yield request
```

Share Improve this answer Follow

Where to set proxy like this in my implementation. Please have a look at question. – [Om Prakash](#) Jan 8, 2018 at 10:43

@OmPrakash check this one and let me know if it helps you [stackoverflow.com/questions/46567024/...](#) – [parik](#) Jan 8, 2018 at 10:48

Start asking to get answers

Find the answer to your question by asking.

Ask question

Explore related questions

python python-3.x scrapy

See similar questions with these tags.

