第 5 章

IP 代理

对于采取了比较强的反爬措施网站来说，要想顺利爬取网站数据，设置随机 User-agent 和 IP 代理是非常有效的两个方法。

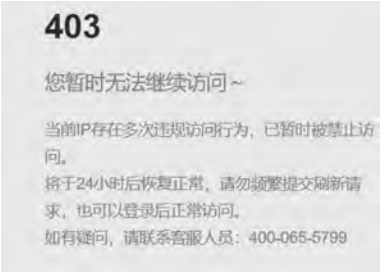

图 5.1　频繁爬取之后

代理就是换个身份，网络中的身份之一就是 IP。在爬虫中，有些网站可能为了防止爬虫或者 DDoS 等，会记录每个 IP 的访问次数。比如，有些网站允许一个 IP 在 1s 内只能访问 10 次等，那么就需要访问一次换一个 IP。如果不对 IP 做任何设置，频繁访问网站之后，网站就可能会出现图 5.1 的提示。使用 IP 代理可以解除被封 IP 这种反爬机制。

IP 代理的获取，可以从以下几个途径得到：

（1）从免费的网站上获取，但质量很低，能用的 IP 极少；

（2）购买收费的代理服务，质量高很多；

（3）自己搭建代理服务器，虽然稳定，但需要大量的服务器资源。

5.1　IP 代理的作用

（1）可以突破自身 IP 的限制，访问一些不许访问的站点；

（2）隐藏自身真实 IP，以免被封锁。

如果想查看本机的 IP，可以有多种方式，下面提供两种简单的查看本机 IP 的方法。

【例 5-1】　查看本机 IP。

方法 1：在百度输入 IP 看到的就是本机 IP，如图 5.2 所示。

图 5.2　查看本机 IP

方法 2：通过程序查看，代码实现如下：

```
import requests
url = 'http://icanhazip.com'
try:
    response = requests.get(url) #不使用代理
    print(response.status_code)
    if response.status_code == 200:
        print(response.text)
except requests.ConnectionError as e:
    print(e.args)
```

程序执行的结果如下：

```
200
123.123.40.xxx #后三位隐去了
```

5.2 IP 代理使用方法

常见的代理包括 HTTP 代理和 SOCKS5 代理，这里主要介绍 HTTP 代理。HTTP 可以在网上搜索一些免费的 IP 代理进行测试。然后在 Requests 请求方法加入一个参数 proxies={字典}实现对 IP 代理的使用，其中字典的键名是 HTTP 或者 HTTPS，键对应的值为一个 IP 地址。

例如 proxies={'http':'59.120.117.244'}，其中键名“http”表示协议类型，键值“59.120.117.244”表示代理。

如果请求 URL 使用的是 HTTP 协议，那么就使用 HTTP 代理；如果请求 URL 使用的是 HTTPS，就使用 HTTPS 代理。

【例 5-2】 设置 IP 代理爬取百度首页，代码如下：

```
import requests
url = "http://www.baidu.com/s"
headers = {
'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/86.0.4240.75 Safari/537.36'  }
response = requests.get(url = url,headers = headers, proxies = {'http':'59.120.117.244'})
with open('baidu.html','w',encoding = 'utf-8')as fp:
    fp.write(response.text)
```

5.3 搭建 IP 池

在爬虫过程中，使用本地 IP 频繁爬取某个网站后，可能会导致 IP 被封，因此有必要搭建一个 IP 池。

【例 5-3】 搭建一个免费的 IP 池。

【解析】 爬取快代理网站上的免费 IP，链接如下：

“https://www.kuaidaili.com/free/”

5.3.1 获取单页 IP

首先定义一个获取目标网页的方法，爬取目标网页如图 5.3 所示，解析出该目标页的 HTTP、PORT、IP。

IP	PORT	匿名度	类型	位置	响应速度	最后验证时间
183.195.106.118	8118	高匿名	HTTP	上海市上海市 移动	7秒	2022-01-07 06:31:01
121.13.252.61	41564	高匿名	HTTP	广东省东莞市 电信	2秒	2022-01-07 05:31:01
222.78.6.70	8083	高匿名	HTTP	中国 福建 漳州 电信	7秒	2022-01-07 04:31:01
14.215.212.37	9168	高匿名	HTTP	中国 广东 东莞 电信	4秒	2022-01-07 03:31:01
111.59.199.58	8118	高匿名	HTTP	中国 广西 百色 移动	3秒	2022-01-07 02:31:01
117.114.149.66	55443	高匿名	HTTP	北京市海淀区 BJENET宽带网络 BGP多线	0.5秒	2022-01-07 01:31:01
27.42.168.46	55481	高匿名	HTTP	中国 广东 中山 联通	2秒	2022-01-07 00:31:01
106.15.197.250	8001	高匿名	HTTP	中国 上海 联通	1秒	2022-01-06 23:31:02
47.243.68.117	8080	高匿名	HTTP	中国 香港 阿里云	1秒	2022-01-06 22:31:01
152.136.62.181	9999	高匿名	HTTP	中国	0.5秒	2022-01-06 21:31:01
152.136.62.181	9999	高匿名	HTTP	中国	0.5秒	2022-01-06 20:31:01
218.75.102.198	8000	高匿名	HTTP	浙江省金华市 电信	0.3秒	2022-01-06 19:31:01
183.247.215.218	30001	高匿名	HTTP	中国 浙江 台州 移动	0.3秒	2022-01-06 18:31:01

图 5.3 待解析的目标网页

操作步骤如下：

第一步：查看 Network 面板，找到目标数据所在请求资源文件的 Headers，如图 5.4 所示，对 Request URL 发送 GET 请求，获得网页数据。

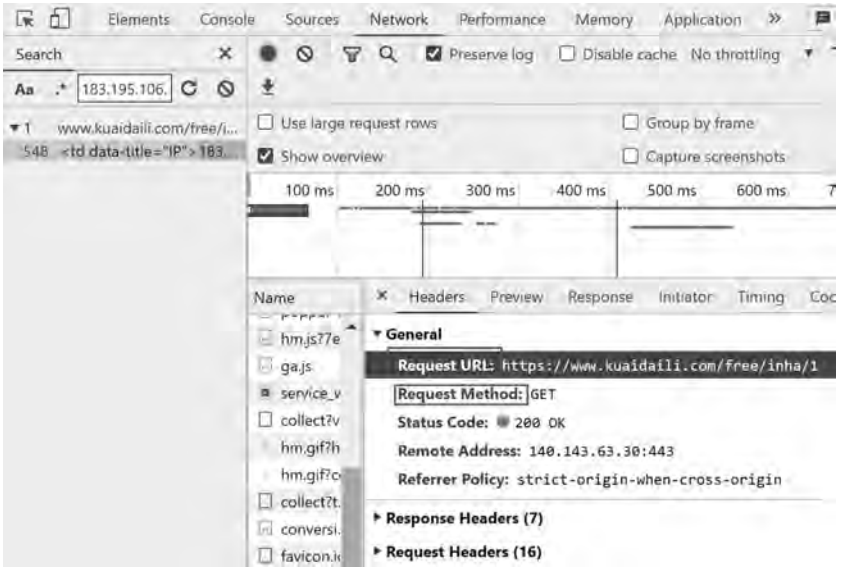

图 5.4 查看目标数据的 Headers

第二步：解析网页数据，获得 PORT、IP 和协议类型。

代码如下：

```
from lxml import etree
def get_content(url):
    headers = {'User-Agent':'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36'}#加载伪装头防止反爬
    response = requests.get(url,headers = headers)
    if response.status_code == 200:
        print('连接正常')
        html = etree.HTML(response.text)
        IP = html.xpath('//td[@data-title = "IP"]/text()')
        PORT = html.xpath('//td[@data-title = "PORT"]/text()')
        TYPE = html.xpath( '//td[@data-title = "类型"]/text()')
        return IP,PORT,TYPE
    else:
        print('连接失败或遭遇反爬')
```

5.3.2 获取多页 IP

对获取的 IP、PORT 和协议类型封装成键值对的字典格式，并存储在字典列表中，已备监测可用性使用，代码实现如下：

```
import time
diclist = []
def get_url(page):                              #传入的数据是爬取目标网页的页数
    for i in range(int(page)):
        try:
            print('正在爬取第%d页'%(i+1))
            url = 'https://www.kuaidaili.com/free/inha/{}/'.format(i+1)
            print("爬取网址为:",url)
            IP,PORT,TYPE = get_content(url)  #这是一个自定义解析网页内容的方法
            for i in range(len(IP)):
                dic = {}
                dic[TYPE[i]] = IP[i] + ':' + PORT[i]
                diclist.append(dic)
            time.sleep(3)                       #防止访问频率过快,被封
        except Exception as e:
            print('爬取失败',e)
```

调用该方法之后程序执行结果如图 5.5 所示。

5.3.3 检测 IP 有效性

利用百度网页进行检测，将 IP 传入 proxies 参数中去验证，如果在规定的时间内返回的 state_code = 200，就说明 IP 是有效的，否则就报错，代码实现如下：

```
def check_ip(reg):#网上检查 IP 的有效性
    url = 'https://www.baidu.com/'
    headers = {'User-Agent':'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36
```

```
正在爬取第1页
爬取网址为： https://www.kuaidaili.com/free/inha/1/
连接正常
正在爬取第2页
爬取网址为： https://www.kuaidaili.com/free/inha/2/
连接正常

[{'HTTP': '113.96.62.246:8081'},
 {'HTTP': '183.195.106.118:8118'},
 {'HTTP': '121.13.252.61:41564'},
 {'HTTP': '222.78.6.70:8083'},
 {'HTTP': '14.215.212.37:9168'},
 {'HTTP': '111.59.199.58:8118'},
 {'HTTP': '117.114.149.66:55443'},
 {'HTTP': '27.42.168.46:55481'},
 {'HTTP': '106.15.197.250:8001'},
 {'HTTP': '47.243.68.117:8080'},
 {'HTTP': '152.136.62.181:9999'},
 {'HTTP': '152.136.62.181:9999'},
 {'HTTP': '218.75.102.198:8000'},
 {'HTTP': '183.247.215.218:30001'},
 {'HTTP': '47.243.190.108:7890'}
```

图 5.5　得到 IP 地址

```
(KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36'}
    can_use = []
    for i in reg:
        try:
            response = requests.get(url,headers,proxies = i,timeout = 1)
            if response.status_code == 200:
                can_use.append(i)
        except Exception as e:
            print('出现问题',e)
    return can_use
```

5.3.4　建立 IP 池

把检测有效的 IP 地址存储之后，就有了一个 IP 池，以备后续在爬取程序的时候，可以随机使用自己建立的 IP 池中的 IP 地址，代码如下：

```
can_ip = check_ip(diclist)
def save_ip(can_ip):
    with open('ip.txt','w + ') as f:
        for i in data:
            f.write(str(i) + '\n')
        f.close()
```

5.4　付费 IP 代理使用

免费的 IP 往往效果不是很好，可以搭建 IP 代理池，但是搭建一个 IP 代理池成本很高，如果只是个人平时偶尔使用一下爬虫，也可以考虑付费 IP，几块钱买个几小时动态 IP，多数情况下都足够爬一个网站了。这里推荐一个付费代理“阿布云”代理。

“阿布云”代理网址：https://www.abuyun.com/http-proxy/dyn-manual-python.html，代码实现如下：

```
import requests
# 要访问的目标页面
targetUrl = "http://test.abuyun.com"
#targetUrl = "http://proxy.abuyun.com/switch-ip"
#targetUrl = "http://proxy.abuyun.com/current-ip"
# 代理服务器
proxyHost = "http-dyn.abuyun.com"
proxyPort = "9020"
# 代理隧道验证信息
proxyUser = "H01234567890123D"
proxyPass = "0123456789012345"
proxyMeta = "http://%(user)s:%(pass)s@%(host)s:%(port)s" % {
    "host" : proxyHost,
    "port" : proxyPort,
    "user" : proxyUser,
    "pass" : proxyPass,
}
proxies = {
    "http" : proxyMeta,
    "https" : proxyMeta,
}
resp = requests.get(targetUrl, proxies=proxies)
print (resp.status_code)
print( resp.text)
```

首次使用的话,可以选择购买一个小时的动态版试用,单击生成隧道代理信息作为凭证加入代码中即可。