Python爬虫IP被限制的解决方法

**引言**

网络爬虫是用于数据采集的一门技术，可以自动高效地进行信息的获取与筛选。从技术手段来说，网络爬虫有多种实现方案，如Python、Java、...。使用Python也会有很多不同的技术方案（urllib、requests、...)，每种技术各有各的特点，只需掌握一种技术，其它便迎刃而解。

但是，网络爬虫的难点并不在于网络爬虫本身，而在于网页的分析与爬虫的反爬攻克问题。高强度、高频率地爬取网页信息常常会给网站的服务器带来巨大压力，因此在利用Python进行网页爬虫的时候，经常会遇到所要爬取的网站采取了反爬虫技术，同一个IP反复爬取一个网站，就很可能被限制IP，无法访问该网站。

本文不再具体介绍Python爬虫和网页分析的技术，网上教程资源很多，主要介绍在使用Python爬虫时遇到的IP被限制的问题，及其简单有效的解决方法。

**反爬虫示例**

以某房产网站为例，爬虫抓取网站内容时会出现以下错误页面，导致爬虫无法继续抓取原始页面数据。



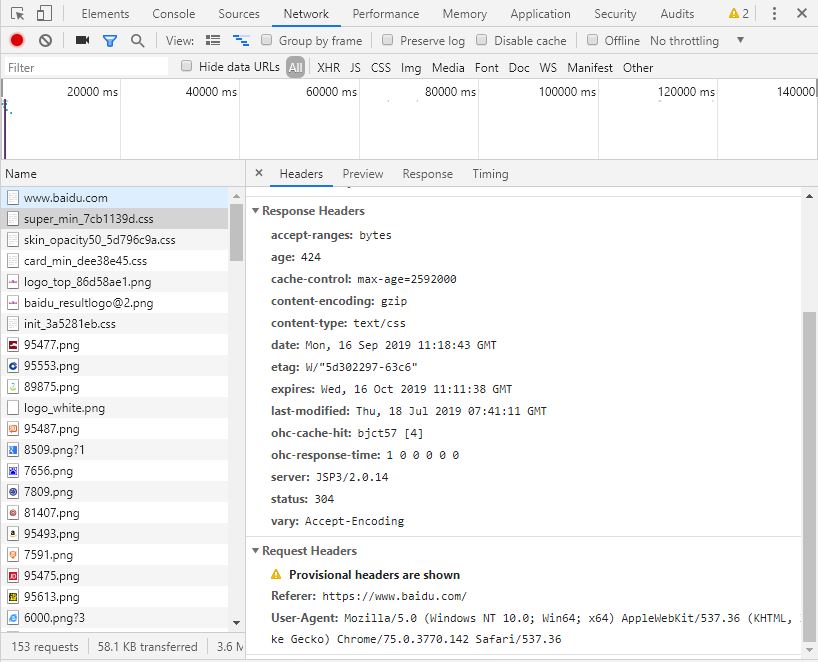
**常用的解决方法**

1. 模拟浏览器访问

网上经常会看到有些爬虫代码会设置headers变量，这是用来模拟浏览器上网用的，因为有些网站设置的反爬虫机制，需要使用浏览器上网才可以访问。

headers的设置也非常简单。

首先，打开调试面板，找到自己浏览器发送请求时的headers，其中，User-agent是用来表示请求者的信息，可以搜集User-agent并保存，爬取过程中动态更换在User-agent，伪装成浏览器的形式。如下图所示。



然后，在爬虫代码中加入headers变量

import requests

url = "<url>"

headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36'}

web\_data = requests.get(url, headers=headers)

这样就能成功的爬取到部分网站的网页信息了。

2. 设置动态IP代理

通过设置模拟浏览器代理来访问获取页面内容，始终只有一个IP地址用来访问,用的多了同样也会被某些网站封掉。

所以设置动态IP代理进行访问成为一种反爬虫的更好策略，即通过其他的IP代替自己的IP进行网站访问。

免费的代理IP网站有很多，虽然这种网站都不会提供太多的代理IP，但足够我们使用了。本文以西刺代理的免费IP网站（www.xicidaili.com/nn/）为例，介绍一下在python中如何使用动态IP代理。

打开网站，可以看到这里有很多代理IP列表，总共超过30万条代理IP地址。



设置IP代理需要修改proxies变量，proxies变量的格式是一个字典：{‘http’: ‘http://xx.xx.xx.xx:xxxx‘}，需要注意的是，proxises有http与https两种，在爬取不同网站时我们需要选用不同类型的网站时选用不同的proxise。

http型：{‘http’: ‘http://42.84.226.65:8888‘}

https型：{‘https’: ‘http://124.193.37.5:8888‘}

然后，从代理IP列表随机抓取IP地址，组成proxies，代码如下。

import random

from bs4 import BeautifulSoup

def get\_ip\_list(url, headers):  
 web\_data = requests.get(url, headers=headers)  
 soup = BeautifulSoup(web\_data.text, 'html.parser')  
 ips = soup.find\_all('tr')  
 ip\_list = []  
 for i in range(1, len(ips)):  
 ip\_info = ips[i]  
 tds = ip\_info.find\_all('td')  
 ip\_list.append(tds[1].text + ':' + tds[2].text)  
 return ip\_list  
  
  
def get\_random\_ip(ip\_list):  
 proxy\_list = []  
 for ip in ip\_list:  
 proxy\_list.append('http://' + ip)  
 proxy\_ip = random.choice(proxy\_list)

# 注意此处修改http或https  
 proxies = {'http': proxy\_ip}  
 return proxies

url = 'http://www.xicidaili.com/nn/'  
headers = {  
 'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36'  
}  
ip\_list = get\_ip\_list(url, headers=headers)  
proxies = get\_random\_ip(ip\_list)

这样，就可以在requests.get()函数中添加proxies参数，设置代理IP。

requests.get(url, headers=headers, proxies=proxies)

通过将获取代理IP的代码放置到合适的位置，实现实时动态的IP代理更新，每次抓取页面都可以获取新的IP，达到防止IP被限制的目的。