

# 通过IP代理池切换IP

## IP代理池

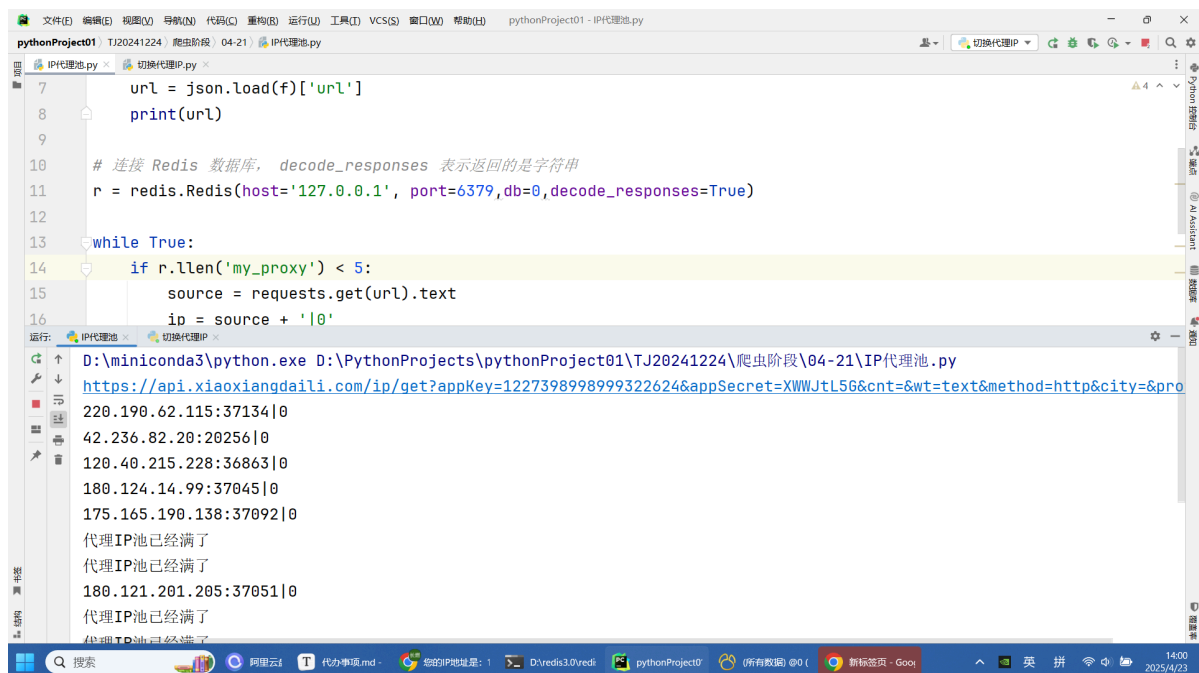
```
import time
import json
import redis
import requests

with open('./url.json', 'r', encoding='utf-8') as f:
    url = json.load(f)['url']
    print(url)

# 连接 Redis 数据库, decode_responses 表示返回的是字符串
r = redis.Redis(host='127.0.0.1', port=6379, db=0, decode_responses=True)

while True:
    if r.llen('my_proxy') < 5:
        source = requests.get(url).text
        ip = source + '|0'
        print(ip)
        r.lpush("my_proxy", ip)
    else:
        time.sleep(2)
        print('代理IP池已经满了')
```

## 运行结果



```
pythonProject01 | TJ20241224 | 爬虫阶段 | 04-21 | IP代理池.py
7      url = json.load(f)['url']
8      print(url)
9
10     # 连接 Redis 数据库, decode_responses 表示返回的是字符串
11     r = redis.Redis(host='127.0.0.1', port=6379, db=0, decode_responses=True)
12
13     while True:
14         if r.llen('my_proxy') < 5:
15             source = requests.get(url).text
16             ip = source + '|0'
17             print(ip)
18             r.lpush("my_proxy", ip)
19         else:
20             time.sleep(2)
21             print('代理IP池已经满了')
```

运行: IP代理池 | 切换代理IP

D:\miniconda3\python.exe D:\PythonProjects\pythonProject01\TJ20241224\爬虫阶段\04-21\IP代理池.py  
https://api.xiaoxiangdaili.com/ip/get?appKey=1227398998999322624&appSecret=XWWJtL56&cnt=&wt=text&method=http&city=&pro  
220.190.62.115:37134|0  
42.236.82.20:20256|0  
120.40.215.228:36863|0  
180.124.14.99:37045|0  
175.165.190.138:37092|0  
代理IP池已经满了  
代理IP池已经满了  
180.121.201.205:37051|0  
代理IP池已经满了  
代理IP池已经满了

## 通过IP代理池切换IP

```
import redis
import requests
from lxml import etree
from loguru import logger
from DrissionPage import ChromiumPage, ChromiumOptions

headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36'
}

r = redis.Redis(host='127.0.0.1', port=6379, db=0, decode_responses=True)

# 获取一组代理IP
def get_ip():
    ip_port_num = r.rpop('my_proxy')
    print(ip_port_num)
    return ip_port_num

# 切换SwitchyOmega代理
def switch_ip(ip_port=None):
    if ip_port:
        # 将代理 IP 字符串拆分为 ip 和 port
        ip, port_number = ip_port.split(":")
        port_number = port_number.split("|")
        # 新开一个标签页进入插件的代理设置页面
        tab = browser.new_tab()
        tab.get("chrome-extension://padekgcemlokbadohgkifijomclgjgif/options.html#!/profile/proxy")
        # 输入 IP 和端口，清除旧值
        tab.ele('x://input[@ng-model="proxyEditors[scheme].host"]').clear().input(ip)
        tab.ele('x://input[@ng-model="proxyEditors[scheme].port"]').clear().input(port)
        # 点击应用按钮，确认设置
        tab.ele('x://a[@ng-click="applyOptions()"]').click()
        # 处理可能弹出的提示框
        # txt = tab.handle_alert()
        # print("提示框", txt)
        # tab.handle_alert(accept=False) # 点击“取消”关闭提示框
    else:
        # 如果传入空代理，恢复为直连
        tab = browser.new_tab()
        tab.get("chrome-extension://padekgcemlokbadohgkifijomclgjgif/popup/index.html#")
        tab.ele('x://span[text()="[直接连接]"').click()
        # 关闭新开的标签页
        if len(browser.tab_ids) > 1:
            print("当前tab个数", len(browser.tab_ids))
            tab.close()

co = ChromiumOptions()
co.set_argument('--start-maximized')
```

```

# 加载 SwitchyOmega 插件扩展（用于切换代理）
co.add_extension(r'D:\SwitchyOmega_Chromium')

# 初始化 Chromium 浏览器对象
browser = ChromiumPage(co)

# 遍历获取到的IP并依次设置代理并检测是否成功
for i in range(10):
    ip_port_num = get_ip()
    ip, port_number = ip_port_num.split(":")
    port, number_str = port_number.split("|")
    number = int(number_str)
    logger.info(f"~~~切换ip, now:{ip}")
    switch_ip(f"{ip}:{port}") # 设置代理
    browser.wait(1) # 等待代理生效
    try:
        proxies = {
            'http': 'http://' + ip + ':' + port,
            'https': 'http://' + ip + ':' + port
        }
        response = requests.get('https://2025.ip138.com/', headers=headers,
proxies=proxies, timeout=5)
        # 从 ip138 获取当前 IP, 验证代理是否生效
        if response.status_code in (200, 302):
            tree = etree.HTML(response.text)
            data = tree.xpath('//p[1]/text()')
            address = data[2].split(':')[1].split(' ')[0]
            logger.success(f">>>>>>>切换代理成功 {address}")
        else:
            if number <= 3:
                number += 1
                r.lpush('my_proxy', ip + ":" + port+ "|" + str(number))
            else:
                continue
    except Exception as err:
        # 如果出错, 记录日志
        logger.error(f"-----切换代理失败 dp {err}")
        browser.wait(10) # 等待10秒以稳定代理使用

# 关闭浏览器
browser.quit()

```

运行结果

