

# SPIDER-DAY03

## 1. lxml解析库

### 1.1 安装使用流程

```
1  【1】安装
2      sudo pip3 install lxml
3
4  【2】使用流程
5      2.1》导模块：          from lxml import etree
6      2.2》创建解析对象：      parse_html = etree.HTML(html)
7      2.3》解析对象调用xpath：r_list = parse_html.xpath('xpath表达式')
```

### 1.2 lxml+xpath使用

```
1  【1】基准xpath：匹配所有电影信息的节点对象列表
2      //dl[@class="board-wrapper"]/dd
3      [<element dd at xxx>,<element dd at xxx>,...]
4
5  【2】遍历对象列表，依次获取每个电影信息
6      item = {}
7      for dd in dd_list:
8          item['name'] = dd.xpath('..//p[@class="name"]/a/text()').strip()
9          item['star'] = dd.xpath('..//p[@class="star"]/text()').strip()
10         item['time'] = dd.xpath('..//p[@class="releasetime"]/text()').strip()
```

## 2. 豆瓣图书爬虫

### 2.1 需求分析

```
1 【1】 抓取目标 - 豆瓣图书top250的图书信息
2     https://book.douban.com/top250?start=0
3     https://book.douban.com/top250?start=25
4     https://book.douban.com/top250?start=50
5     ... ..
6
7 【2】 抓取数据
8     2.1) 书籍名称 : 红楼梦
9     2.2) 书籍描述 : [清] 曹雪芹 著 / 人民文学出版社 / 1996-12 / 59.70元
10    2.3) 书籍评分 : 9.6
11    2.4) 评价人数 : 286382人评价
12    2.5) 书籍类型 : 都云作者痴, 谁解其中味?
```

## 2.2 实现流程

```
1 【1】 确认数据来源 - 响应内容存在
2 【2】 分析URL地址规律 - start为0 25 50 75 ...
3 【3】 xpath表达式
4     3.1) 基准xpath, 匹配每本书籍的节点对象列表
5         //div[@class="indent"]/table
6
7     3.2) 依次遍历每本书籍的节点对象, 提取具体书籍数据
8         书籍名称 : .//div[@class="p12"]/a/@title
9         书籍描述 : .//p[@class="p1"]/text()
10        书籍评分 : .//span[@class="rating_nums"]/text()
11        评价人数 : .//span[@class="p1"]/text()
12        书籍类型 : .//span[@class="inq"]/text()
```

## 2.3 代码实现

```
1 import requests
2 from lxml import etree
3 import time
4 import random
5 from fake_useragent import UserAgent
6
7 class DoubanBookSpider:
8     def __init__(self):
9         self.url = 'https://book.douban.com/top250?start={}'
10
11    def get_html(self, url):
12        headers = { 'User-Agent':UserAgent().random }
13        html = requests.get(url=url, headers=headers).content.decode('utf-8', 'ignore')
14        # 直接调用解析函数
15        self.parse_html(html)
16
17    def parse_html(self, html):
18        p = etree.HTML(html)
```

```

19     # 基准xpath,匹配每本书的节点对象列表
20     table_list = p.xpath('//div[@class="indent"]/table')
21     for table in table_list:
22         item = {}
23         # 书名
24         name_list = table.xpath('.//div[@class="pl2"]/a/@title')
25         item['book_name'] = name_list[0].strip() if name_list else None
26         # 信息
27         info_list = table.xpath('.//p[@class="pl1"]/text()')
28         item['book_info'] = info_list[0].strip() if info_list else None
29         # 评分
30         score_list = table.xpath('.//span[@class="rating_nums"]/text()')
31         item['book_score'] = score_list[0].strip() if score_list else None
32         # 人数
33         number_list = table.xpath('.//span[@class="pl1"]/text()')
34         item['book_number'] = number_list[0].strip()[1:-1].strip() if number_list else
None
35         # 描述
36         comment_list = table.xpath('.//span[@class="inq"]/text()')
37         item['book_comment'] = comment_list[0].strip() if comment_list else None
38
39         print(item)
40
41     def run(self):
42         for i in range(10):
43             start = i * 25
44             page_url = self.url.format(start)
45             self.get_html(url=page_url)
46             # 控制数据抓取的频率,uniform生成指定范围内浮点数
47             time.sleep(random.uniform(0, 3))
48
49
50 if __name__ == '__main__':
51     spider = DoubanBookSpider()
52     spider.run()

```

## 3. 百度贴吧小视频爬虫

### 3.1 需求分析

- 1 【1】官网地址：进入某个百度贴吧，寻找有视频的帖子，比如如下帖子链接：
- 2 <https://tieba.baidu.com/p/7185877941>
- 3
- 4 【2】目标
- 5 2.1> 在此帖子中提取中具体视频的链接
- 6 2.2> 将视频抓取保存到本地文件

### 3.2 实现流程

```
1  【1】确认数据来源：静态!!!
2  【2】帖子中视频的xpath表达式
3
4  ### 重要：页面中xpath不能全信，一切以响应内容为主
5  ### 重要：页面中xpath不能全信，一切以响应内容为主
6  ### 重要：页面中xpath不能全信，一切以响应内容为主
7  ### 重要：页面中xpath不能全信，一切以响应内容为主
8  ### 重要：页面中xpath不能全信，一切以响应内容为主
9  ### 重要：页面中xpath不能全信，一切以响应内容为主
10 ### 重要：页面中xpath不能全信，一切以响应内容为主
11 ### 重要：页面中xpath不能全信，一切以响应内容为主
12 ### 重要：页面中xpath不能全信，一切以响应内容为主
13 ### 重要：页面中xpath不能全信，一切以响应内容为主
14 ### 重要：页面中xpath不能全信，一切以响应内容为主
```

### 3.3 代码实现

```
1  import requests
2  from lxml import etree
3
4  # 向帖子链接发请求
5  url = 'https://tieba.baidu.com/p/7185877941'
6  headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
7  like Gecko) Chrome/87.0.4280.88 Safari/537.36'}
8  html = requests.get(url=url, headers=headers).text
9
10 # 视频链接的xpath表达式 - 一切以响应内容为准
11 x = '//div[@class="video_src_wrapper"]/embed/@data-video'
12 eobj = etree.HTML(html)
13 video_url_list = eobj.xpath(x)
14
15 # 将视频抓取保存到本地文件
16 if video_url_list:
17     video_url = video_url_list[0]
18     video_html = requests.get(url=video_url, headers=headers).content
19     with open('girl.mp4', 'wb') as f:
20         f.write(video_html)
21 else:
22     print('提取视频链接失败')
```

## 4. 代理参数

### 4.1 代理IP概述

```
1  【1】 定义
2      代替你原来的IP地址去对接网络的IP地址
3
4  【2】 作用
5      隐藏自身真实IP,避免被封
6
7  【3】 获取代理IP网站
8      快代理、全网代理、代理精灵、... ...
9
10 【4】 参数类型
11     proxies
12     proxies = { '协议': '协议://IP:端口号' }
13     proxies = { '协议': '协议://用户名:密码@IP:端口号' }
```

## 4.2 代理分类

### 4.2.1 普通代理

```
1  【1】 代理格式
2      proxies = { '协议': '协议://IP:端口号' }
3
4  【2】 使用免费普通代理IP访问测试网站: http://httpbin.org/get
5
6  import requests
7  url = 'http://httpbin.org/get'
8  headers = {'User-Agent': 'Mozilla/5.0'}
9  # 定义代理,在代理IP网站中查找免费代理IP
10 proxies = {
11     'http': 'http://112.85.164.220:9999',
12     'https': 'https://112.85.164.220:9999'
13 }
14 html = requests.get(url,proxies=proxies,headers=headers,timeout=5).text
15 print(html)
```

### 4.2.2 私密代理和独享代理

```
1  【1】 代理格式
2      proxies = { '协议': '协议://用户名:密码@IP:端口号' }
3
4  【2】 使用私密代理或独享代理IP访问测试网站: http://httpbin.org/get
5
6  import requests
7  url = 'http://httpbin.org/get'
8  proxies = {
9      'http': 'http://309435365:szayclhp@106.75.71.140:16816',
10     'https': 'https://309435365:szayclhp@106.75.71.140:16816',
11 }
12 headers = {
13     'User-Agent' : 'Mozilla/5.0',
14 }
```

```
15
16 html = requests.get(url,proxies=proxies,headers=headers,timeout=5).text
17 print(html)
```

## 4.3 建立代理IP池

```
1 """
2 建立开放代理的代理ip池
3 """
4 import requests
5
6 class ProxyPool:
7     def __init__(self):
8         self.api_url = 'http://dev.kdlapi.com/api/getproxy/?
9         orderid=999955248138592&num=20&protocol=2&method=2&an_ha=1&sep=1'
10        self.test_url = 'http://httpbin.org/get'
11        self.headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
12        AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.83 Safari/537.36'}
13
14    def get_proxy(self):
15        html = requests.get(url=self.api_url, headers=self.headers).text
16        # proxy_list: ['1.1.1.1:8888', '2.2.2.2:9999,...]
17        proxy_list = html.split('\r\n')
18        for proxy in proxy_list:
19            # 测试proxy是否可用
20            self.test_proxy(proxy)
21
22    def test_proxy(self, proxy):
23        """测试1个代理ip是否可用"""
24        proxies = {
25            'http' : 'http://{0}'.format(proxy),
26            'https': 'https://{0}'.format(proxy),
27        }
28        try:
29            resp = requests.get(url=self.test_url, proxies=proxies, headers=self.headers,
30            timeout=3)
31            if resp.status_code == 200:
32                print(proxy, '\033[31m可用\033[0m')
33            else:
34                print(proxy, '不可用')
35        except Exception as e:
36            print(proxy, '不可用')
37
38    def run(self):
39        self.get_proxy()
40
41 if __name__ == '__main__':
42     spider = ProxyPool()
43     spider.run()
```

## 5. requests.post()

### 5.1 POST请求

- 1 【1】适用场景：Post类型请求的网站
- 2
- 3 【2】参数：data={}
- 4     2.1) Form表单数据：字典
- 5     2.2) res = requests.post(url=url, data=data, headers=headers)
- 6
- 7 【3】POST请求特点：Form表单提交数据

### 5.2 控制台抓包

#### ■ 打开方式及常用选项

- 1 【1】打开浏览器，F12打开控制台，找到Network选项卡
- 2
- 3 【2】控制台常用选项
- 4     2.1) Network：抓取网络数据包
- 5         a> ALL：抓取所有的网络数据包
- 6         b> XHR：抓取异步加载的网络数据包
- 7         c> JS：抓取所有的JS文件
- 8     2.2) Sources：格式化输出并打断点调试JavaScript代码，助于分析爬虫中一些参数
- 9     2.3) Console：交互模式，可对JavaScript中的代码进行测试
- 10
- 11 【3】抓取具体网络数据包后
- 12     3.1) 单击左侧网络数据包地址，进入数据包详情，查看右侧
- 13     3.2) 右侧：
- 14         a> Headers：整个请求信息
- 15             General、Response Headers、Request Headers、Query String、Form Data
- 16         b> Preview：对响应内容进行预览
- 17         c> Response：响应内容

## 6. 今日作业

```
1  【1】链家二手房爬虫
2      1.1> 官网地址：进入链家官网，点击二手房：https://bj.lianjia.com/ershoufang/
3      1.2> 目标：抓取100页的二手房源信息，包含房源的：
4          名称
5          地址
6          户型、面积、方位、是否精装、楼层、年代、类型
7          总价
8          单价
9      1.3> 数据处理
10         将数据分别存入：MySQL、MongoDB、csv文件中
11
12  【2】抓取快代理网站免费高匿代理，并测试是否可用来建立自己的代理IP池
13      https://www.kuaidaili.com/free/
```