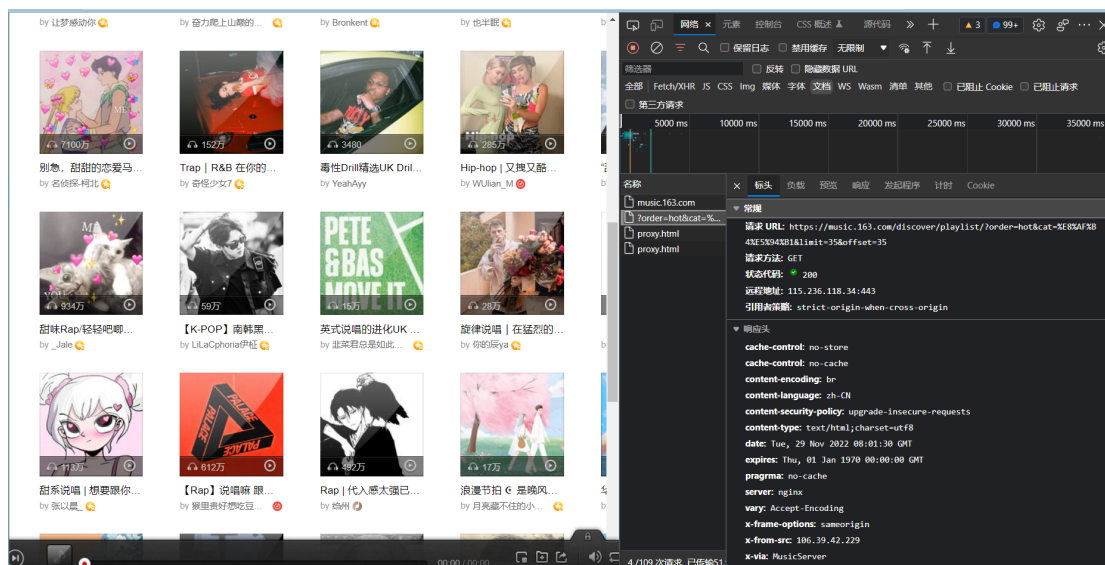
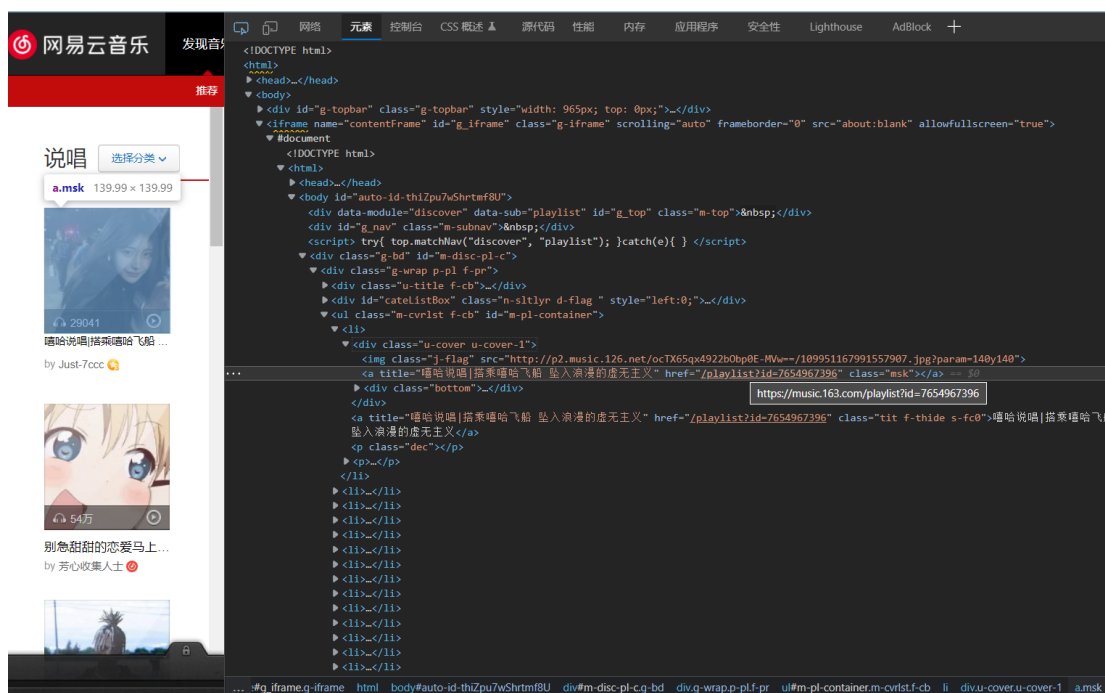


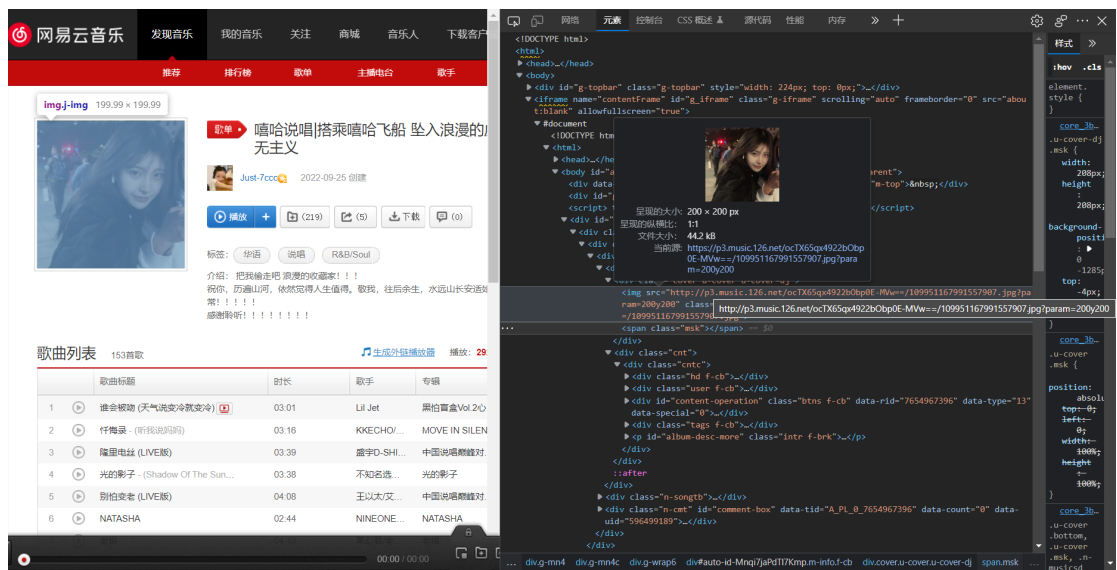
1 网页解析



解析网页，在【文档】类型中找到要爬取的目标 URL。



解析元素，找到目标歌单的 URL 所在位置



解析子 URL，找到想要爬取的内容的位置。

2 代码

2.1 准备工作

```
1 import pandas as pd
2 import requests, time, csv, random, re, pickle
3 from lxml import etree
4 from threading import Thread, Lock
5 from queue import Queue
6
7
8 # 将网页参数设置为全局变量以便使用
9 agents = ["Mozilla/5.0 (Android; Mobile; rv:14.0) Gecko/14.0 Firefox/14.0", # 设置多个Agent应对反爬
10           "Mozilla/5.0 (Android; Tablet; rv:14.0) Gecko/14.0 Firefox/14.0",
11           "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; rv:21.0) Gecko/20100101 Firefox/21.0",
12           "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:21.0) Gecko/20130331 Firefox/21.0",
13           "Mozilla/5.0 (Windows NT 6.2; WOW64; rv:21.0) Gecko/20100101 Firefox/21.0",
14           "Mozilla/5.0 (Linux; Android 4.1.1; Nexus 7 Build/JRO03D) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.166 Safari/535.19",
15           "Mozilla/5.0 (Linux; Android 4.0.4; Galaxy Nexus Build/IMM76B) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.133 Mobile",
16           "Mozilla/5.0 (Linux; Android 4.1.2; Nexus 7 Build/J2054K) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.166 Safari/535.19",
17           "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.93 Safari/537.36",
18           "Mozilla/5.0 (compatible; WOW64; MSIE 10.0; Windows NT 6.2)",
19           "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)",
20           "Opera/9.80 (Windows NT 6.1; WOW64; U; en) Presto/2.10.229 Version/11.62"]
21
22 headers = {
23     'Referer': 'https://music.163.com/',
24     'User-Agent': random.choice(agents),
25 }
26
27 def urls_extracting():
28     pl_urls = []
29     for page in range(0, 44):
30         time.sleep(random.uniform(2, 5)) # 设置随机间隔时间反爬
31         #print(f'---开始爬取第{page+1}页--->>>')
32         url = f'https://music.163.com/discover/playlist/?order=hot&cat=8%E8%AF%B4%E5%94%B1&limit=35&offset={page*35}'
33         response = requests.get(url=url, headers=headers).text # 响应内容
34         html = etree.HTML(response) # 转换成XML
35         playlist = html.xpath('//div[@class="u-cover u-cover-1"]/@[class="msk"]/@href') # 获取各歌单的子url列表
36         for i in playlist:
37             pl_url = f'https://music.163.com/' + i # 歌单完整url
38             pl_urls.append(pl_url)
39             #print(f'---第{page+1}页歌单url提取成功---v')
40         with open('week12_pl_urls.csv', 'w', encoding='utf-8', newline='') as f:
41             for i in pl_urls: # 保存pl_urls到本地方便使用
42                 csv.writer(f).writerow([i])
43
44 def write(row): # 写入csv
45     with open(r'week12_网易云歌单.csv', 'a', encoding='utf-8', newline='') as f:
46         csv.writer(f).writerow(row)
```

2.2 生产者尝试连接网页

```
48 def produce(q): # 生产者：网页请求
49     with open('week12_pl_urls.csv', 'r', encoding = 'utf-8', newline = '') as f:
50         pl_urls = [i[0] for i in csv.reader(f)]
51     for i in range(0, len(pl_urls)):
52         #time.sleep(random.uniform(2, 5)) # 设置随机间隔时间反爬
53         print(f'---开始爬取第{i+1}个歌单-->>>')
54         pl_response = requests.get(url=pl_urls[i], headers=headers).text
55         q.put([i, pl_response])
```

2.3 消费者解析网页内容并写入 csv

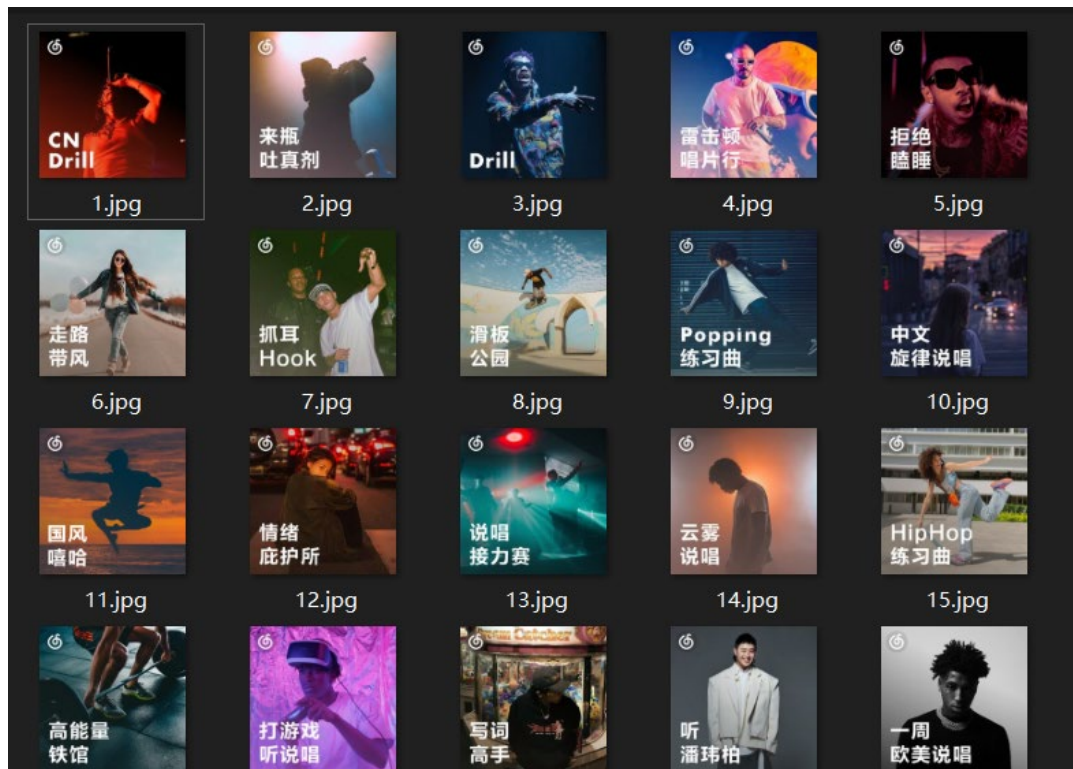
```
57 def consume(q, lock): # 消费者：网页提取
58     while True:
59         item = q.get()
60         if item is None:
61             break
62         i, pl_response = item
63         pl_html = etree.HTML(pl_response)
64         title = pl_html.xpath('//h2[@class="f-ff2 f-brk"]')[0].text
65         cover = pl_html.xpath('//img[@class="j-img"]/@src')[0]
66         with open('week12_网易云歌单封面/'+f'{i+1}'+'.jpg', 'wb') as image:
67             image.write(requests.get(cover).content)
68         creator_name = pl_html.xpath('//a[@class="s-fc7"]')[0].text
69         creator_id = re.search('(?!<=?)id=).*', pl_html.xpath('//a[@class="s-fc7"]/@href')[0])
70         intro_result = pl_html.xpath('//p[@class="intr f-brk"]')
71         if intro_result == []:
72             introduction = ''
73         else:
74             introduction = intro_result[0].xpath('string(.)')[4:] # 删去'介绍\n'
75         num_songs = pl_html.xpath('//span[@id="playlist-track-count"]')[0].text
76         num_play = pl_html.xpath('//strong[@id="play-count"]')[0].text
77         num_add = pl_html.xpath('//a[@class="u-btni u-btni-fav "]/i/text()')[0][1:-1]
78         num_share = pl_html.xpath('//a[@class="u-btni u-btni-share "]/i/text()')[0][1:-1]
79         num_comment = pl_html.xpath('//span[@id="cnt_comment_count"]/text()')[0]
80         row = [title, cover, creator_name, creator_id, introduction,
81               num_songs, num_play, num_add, num_share, num_comment]
82         lock.acquire() # 加锁以免出错
83         try:
84             write(row)
85         finally:
86             lock.release()
87         print(f'>>>--第{i+1}个歌单写入成功--V')
88         q.task_done()
```

2.4 主函数

```
91 if __name__ == '__main__':
92     #urls_extracting()
93     with open(r'week12_网易云歌单.csv', 'a', encoding = 'utf-8', newline = '') as f:
94         csv.writer(f).writerow(['title', 'cover', 'creator_name', 'creator_id', 'introduction',
95                                 'num_songs', 'num_play', 'num_add', 'num_share', 'num_comment'])
96     n = 3 # 消费者线程数
97     lock = Lock()
98     q = Queue()
99     threads = []
100    for i in range(n): # 创建并启动消费者进程
101        t = Thread(target = consume, args = (q, lock))
102        threads.append(t)
103        t.start()
104    produce(q) # 启动生产者线程
105    q.join()
106    for i in range(n): # 停止消费者进程
107        q.put(None)
108    for t in threads:
109        t.join()
```

3 运行结果

```
---开始爬取第1511个歌单-->>>
---开始爬取第1512个歌单-->>>
>>>--第1510个歌单写入成功--√
>>>--第1511个歌单写入成功--√
---开始爬取第1513个歌单-->>>
>>>--第1512个歌单写入成功--√
---开始爬取第1514个歌单-->>>
---开始爬取第1515个歌单-->>>
>>>--第1513个歌单写入成功--√
>>>--第1514个歌单写入成功--√
---开始爬取第1516个歌单-->>>
>>>--第1515个歌单写入成功--√
>>>--第1516个歌单写入成功--√
```



```
week12_网易云歌单.csv - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
title,cover,creator_name,creator_id,introduction,num_songs,num_play,num_add,num_sh
[CN Drill] 中文钻头节奏, 听就完了,http://p4.music.126.net/kmcRWBitc6vvyC-2sE0wvA=
",67,130249,586,7,9
[来瓶吐真剂] 让歌里的故事诉说心事,http://p3.music.126.net/KuhEhth3Dzwu1yYa--Nx6w=
",71,357855,979,13,10
[Drill] 燥起来! 接受钻头说唱的低音轰炸吧,http://p4.music.126.net/M9XVRxHkBFYsAZmO
",60,224669,1610,15,8
[雷击顿唱片行] 让你扭动身体的异域节奏,http://p3.music.126.net/BSs2iYeiFUJclFSiAPP8Bg
",30,2494064,12425,127,25
[拒绝瞌睡] 跟着节拍抖腿的嘻哈音乐,http://p3.music.126.net/AEjvfbLi56fDGcSXQnkxrg=
戴上耳机 我就是这条gai最酷的崽
",30,12251261,82321,658,163
[走路带风] 用音乐给今天充满电,http://p4.music.126.net/hFMcFqf2BPCxVer_ehx0FA==/10
无聊的一天, 如坐针毡, 深陷996苦海...
按下播放键, 一秒带你找回状态!
",100,1243239,5341,45,18
[抓耳Hook] 旋律中邂逅, 把你心牵走,http://p3.music.126.net/l1irmc3FAACneNRuQUView
",70,627940,3544,37,11
```

4 爬虫程序往往需要稳定运行较长的时间，因此如果你的程序突然中断或异常（比如网络或被封），如何能够快速从断点重启？

记录程序运行点，比如记录当前爬虫的页数，在程序突然中断时，从中断的页数重新爬取。