

练习1 - 电影天堂二级页面抓取

领取任务

```
1 # 地址
2 电影天堂 - 2019年新片精品 - 更多
3 # 目标
4 电影名称、下载链接
5
6 # 分析
7 *****一级页面需抓取*****
8     1、电影详情页链接
9
10 *****二级页面需抓取*****
11     1、电影名称
12     2、电影下载链接
```

实现步骤

- 1、确定响应内容中是否存在所需抓取数据
- 2、找URL规律

```
1 第1页 : https://www.dytt8.net/html/gndy/dyzz/list_23_1.html
2 第2页 : https://www.dytt8.net/html/gndy/dyzz/list_23_2.html
3 第n页 : https://www.dytt8.net/html/gndy/dyzz/list_23_n.html
```

- 3、写正则表达式

```
1 1、一级页面正则表达式
2 <table width="100%".*?<td width="5%".*?<a href="(.*?)".*?</td>.*?</table>
3 2、二级页面正则表达式
4 <div class="title_all"><h1><font color=#07519a>(.*?)</font></h1></div>.*?<td style="WORD-
  WRAP.*?>.*?>(.*?)</a>
```

- 4、代码实现

```
1 from urllib import request
2 import re
3 from useragents import ua_list
4 import time
5 import random
6
7 class FilmSkySpider(object):
8     def __init__(self):
9         # 一级页面url地址
```

```

10     self.url = 'https://www.dytt8.net/html/gndy/dyzz/list_23_{}.html'
11
12 # 获取html功能函数
13 def get_html(self,url):
14     headers = {
15         'User-Agent':random.choice(ua_list)
16     }
17     req = request.Request(url=url,headers=headers)
18     res = request.urlopen(req)
19     # 通过网站查看网页源码,查看网站charset='gb2312'
20     # 如果遇到解码错误,识别不了一些字符,则 ignore 忽略掉
21     html = res.read().decode('gb2312','ignore')
22
23     return html
24
25 # 正则解析功能函数
26 def re_func(self,re_bds,html):
27     pattern = re.compile(re_bds,re.S)
28     r_list = pattern.findall(html)
29
30     return r_list
31
32 # 获取数据函数 - html是一级页面响应内容
33 def parse_page(self,one_url):
34     html = self.get_html(one_url)
35     re_bds = r'<table width="100%" .*?<td width="5%" .*?<a href="(.*?)".*?<ulink">.*?</table>'
36     # one_page_list: ['/html/xxx','/html/xxx','/html/xxx']
37     one_page_list = self.re_func(re_bds,html)
38
39     for href in one_page_list:
40         two_url = 'https://www.dytt8.net' + href
41         self.parse_two_page(two_url)
42         # uniform: 浮点数,爬取1个电影信息后sleep
43         time.sleep(random.uniform(1, 3))
44
45
46 # 解析二级页面数据
47 def parse_two_page(self,two_url):
48     item = {}
49     html = self.get_html(two_url)
50     re_bds = r'<div class="title_all"><h1><font color=#07519a>(.*?)</font></h1></div>.*?<td
51 style="WORD-WRAP.*?>.*?>(.*?)</a>'
52     # two_page_list: [('名称1','ftp://xxxx.mkv')]
53     two_page_list = self.re_func(re_bds,html)
54
55     item['name'] = two_page_list[0][0].strip()
56     item['download'] = two_page_list[0][1].strip()
57
58     print(item)
59
60 def main(self):
61     for page in range(1,201):
62         one_url = self.url.format(page)
63         self.parse_page(one_url)
64         # uniform: 浮点数
65         time.sleep(random.uniform(1,3))

```

```

66
67 if __name__ == '__main__':
68     spider = FilmSkySpider()
69     spider.main()

```

■ 5、练习

把电影天堂数据存入MySQL数据库 - 增量爬取

```

1  # 思路
2  # 1、MySQL中新建表 urltab,存储所有爬取过的链接的指纹
3  # 2、在爬取之前,先判断该指纹是否爬取过,如果爬取过,则不再继续爬取

```

练习代码实现

```

1  # 建库建表
2  create database filmskydb charset utf8;
3  use filmskydb;
4  create table request_finger(
5  finger char(32)
6  )charset=utf8;
7  create table filmtab(
8  name varchar(200),
9  download varchar(500)
10 )charset=utf8;

```

```

1  from urllib import request
2  import re
3  from useragents import ua_list
4  import time
5  import random
6  import pymysql
7  from hashlib import md5
8  import sys
9
10 class FilmSkySpider(object):
11     def __init__(self):
12         # 一级页面url地址
13         self.url = 'https://www.dytt8.net/html/gndy/dyzz/list_23_{}.html'
14         self.db = pymysql.connect('localhost','root','attack','filmskydb',charset='utf8')
15         self.cursor = self.db.cursor()
16
17     # 获取html功能函数
18     def get_html(self,url):
19         headers = {
20             'User-Agent':random.choice(ua_list)
21         }
22         req = request.Request(url=url,headers=headers)
23         res = request.urlopen(req)
24         # 通过网站查看网页源码,查看网站charset='gb2312'
25         # 如果遇到解码错误,识别不了一些字符,则 ignore 忽略掉
26         html = res.read().decode('gb2312','ignore')
27
28         return html
29

```

```

30 # 正则解析功能函数
31 def re_func(self, re_bds, html):
32     pattern = re.compile(re_bds, re.S)
33     r_list = pattern.findall(html)
34
35     return r_list
36
37 # 获取数据函数 - html是一级页面响应内容
38 def parse_page(self, one_url):
39     html = self.get_html(one_url)
40     re_bds = r'<table width="100%" .*?<td width="5%" .*?<a href="(.*?)".*?</table>'
41     # one_page_list: ['/html/xxx', '/html/xxx', '/html/xxx']
42     one_page_list = self.re_func(re_bds, html)
43
44     for href in one_page_list:
45         two_url = 'https://www.dytt8.net' + href
46         # 判断在数据库中是否存在此链接, 一旦存在, 直接break, 新更新的链接都在上面
47         sel = 'select finger from request_finger where finger=%s'
48         s = md5()
49         s.update(two_url.encode())
50         finger = s.hexdigest()
51         result = self.cursor.execute(sel, [finger])
52         if not result:
53             self.parse_two_page(two_url)
54             # uniform: 浮点数, 爬取1个电影信息后sleep
55             time.sleep(random.uniform(1, 3))
56             ins = 'insert into request_finger values(%s)'
57             self.cursor.execute(ins, [finger])
58             self.db.commit()
59         else:
60             sys.exit('未更新')
61
62 # 解析二级页面数据
63 def parse_two_page(self, two_url):
64     item = {}
65     html = self.get_html(two_url)
66     re_bds = r'<div class="title_all"><h1><font color=#07519a>(.*?)</font></h1></div>.*?<td style="WORD-WRAP.*?>.*?>(.*?)</a>'
67     # two_page_list: [('名称1', 'ftp://xxxx.mkv')]
68     two_page_list = self.re_func(re_bds, html)
69
70     item['name'] = two_page_list[0][0].strip()
71     item['download'] = two_page_list[0][1].strip()
72     ins = 'insert into filmtab values(%s,%s)'
73     film_list = [
74         item['name'], item['download']
75     ]
76     self.cursor.execute(ins, film_list)
77     self.db.commit()
78     print(film_list)
79
80
81 def run(self):
82     for page in range(1, 201):
83         one_url = self.url.format(page)
84         self.parse_page(one_url)

```

```
86         # uniform: 浮点数
87         time.sleep(random.uniform(1,3))
88
89     if __name__ == '__main__':
90         spider = FilmSkySpider()
91         spider.run()
```

练习2 - 4567tv数据抓取

■ 领取任务

```
1  # 1、爬取地址
2  https://www.4567tv.tv/ --> 动作片
3
4
5  # 2、爬取目标
6  电影名称、电影简介
7
8  # 3、爬取分析
9  *****一级页面需抓取*****
10 1、电影详情页的链接
11
12 *****二级页面需抓取*****
13 1、电影名称
14 2、电影简介
```

■ 实现步骤

```
1  # 1. 确定响应内容中是否存在所需抓取数据 - 存在
2  # 2. 找URL地址规律
3  第1页: https://www.4567tv.tv/index.php/vod/show/id/5/page/1.html
4  第2页: https://www.4567tv.tv/index.php/vod/show/id/5/page/2.html
5  第n页: https://www.4567tv.tv/index.php/vod/show/id/5/page/3.html
6
7  # 3. 写正则表达式
8  一级页面正则:
9  <li class="col-md-6 col-sm-4 col-xs-3">.*?<a class="stui-vodlist__thumb lazyload" href="
10  (.*)">.*?</li>
11
12  二级页面正则:
13  <div class="stui-content__detail">.*?<h1 class="title">(.*?)</h1>.*?<span class="detail-
14  content" style="display: none;">(.*?)</span>
15
16  # 4. 代码实现
```

■ 代码实现

```
1 import requests
2 import re
```

```

3 import time
4 import random
5 from fake_useragent import UserAgent
6
7 class TvSpider(object):
8     def __init__(self):
9         self.url = 'https://www.4567tv.tv/index.php/vod/show/id/5/page/{}.html'
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')
14         return html
15
16     def regex_func(self,regex,html):
17         pattern = re.compile(regex,re.S)
18         r_list = pattern.findall(html)
19         return r_list
20
21     def parse_html(self,one_url):
22         one_html = self.get_html(one_url)
23         one_regex = '<li class="col-md-6 col-sm-4 col-xs-3">.*?<a class="stui-vodlist__thumb lazyload" href="(.*?)".*?</li>'
24         href_list = self.regex_func(one_regex,one_html)
25         for href in href_list:
26             two_link = 'https://www.4567tv.tv' + href
27             self.get_data(two_link)
28             time.sleep(random.uniform(0,1))
29
30     def get_data(self,two_link):
31         two_html = self.get_html(two_link)
32         two_regex = '<div class="stui-content__detail">.*?<h1 class="title">(.*?)</h1>.*?<span class="detail-content" style="display: none;">(.*?)</span>'
33         film_list = self.regex_func(two_regex,two_html)
34         item = {}
35         item['film_name'] = film_list[0][0]
36         item['film_content'] = film_list[0][1]
37
38         print(item)
39
40     def run(self):
41         for i in range(1,11):
42             one_url = self.url.format(i)
43             self.parse_html(one_url)
44
45 if __name__ == '__main__':
46     spider = TvSpider()
47     spider.run()

```

■ 扩展 - 增量爬取

```

1 将数据存入MySQL数据库 - 增量爬取
2
3 # 思路
4 1、MySQL中新建表 urltab,存储所有爬取过的链接的指纹
5 2、在爬取之前,先判断该指纹是否爬取过,如果爬取过,则不再继续爬取
6

```

```

7 # 建库建表
8 create database tvdb charset utf8;
9 use tvdb;
10 create table request_finger(
11 finger char(32)
12 )charset=utf8;
13 create table tvtab(
14 name varchar(100),
15 comment varchar(1000)
16 )charset=utf8;

```

■ 增量爬取 - MySQL

```

1 import requests
2 import re
3 import time
4 import random
5 from fake_useragent import UserAgent
6 import pymysql
7 from hashlib import md5
8 import sys
9
10 class TvSpider(object):
11     def __init__(self):
12         self.url = 'https://www.4567tv.tv/index.php/vod/show/id/5/page/{0}.html'
13         self.db = pymysql.connect('localhost', 'root', '123456', 'tvdb', charset='utf8')
14         self.cursor = self.db.cursor()
15
16     def get_html(self, url):
17         """功能函数1 - 获取相应内容"""
18         headers = {'User-Agent': UserAgent().random}
19         html = requests.get(url=url, headers=headers).content.decode('utf-8')
20         return html
21
22     def regex_func(self, regex, html):
23         """功能函数2 - 正则解析函数"""
24         pattern = re.compile(regex, re.S)
25         r_list = pattern.findall(html)
26         return r_list
27
28     def parse_html(self, one_url):
29         """数据提取函数"""
30         one_html = self.get_html(one_url)
31         one_regex = '<li class="col-md-6 col-sm-4 col-xs-3">.*?<a class="stui-vodlist__thumb lazyload" href="(.*?)".*?</li>'
32         href_list = self.regex_func(one_regex, one_html)
33         for href in href_list:
34             two_link = 'https://www.4567tv.tv' + href
35             # 对链接进行md5加密
36             finger = md5(two_link.encode()).hexdigest()
37             sel = 'select finger from request_finger where finger=%s'
38             result = self.cursor.execute(sel, [finger])
39             if not result:
40                 self.get_data(two_link)
41                 time.sleep(random.uniform(0, 1))
42             # 抓取完成后千万不要忘记存入指纹

```

```

43         ins = 'insert into request_finger values(%s)'
44         self.cursor.execute(ins, [finger])
45         self.db.commit()
46     else:
47         sys.exit('网站未更新数据')
48
49     def get_data(self, two_link):
50         two_html = self.get_html(two_link)
51         two_regex = '<div class="stui-content__detail">.*?<h1 class="title">(.*?)</h1>.*?<span class="detail-content" style="display: none;">(.*?)</span>'
52         film_list = self.regex_func(two_regex, two_html)
53
54         film_name = film_list[0][0]
55         film_content = film_list[0][1]
56         ins = 'insert into tvtab values(%s,%s)'
57         self.cursor.execute(ins, [film_name, film_content])
58         self.db.commit()
59         print(film_name, film_content)
60
61     def run(self):
62         for i in range(1, 11):
63             one_url = self.url.format(i)
64             self.parse_html(one_url)
65
66 if __name__ == '__main__':
67     spider = TvSpider()
68     spider.run()

```

■ 能不能使用redis来实现增量

```

1  """
2      提示：使用redis中的集合,sadd()方法,添加成功返回1,否则返回0
3      请各位大佬忽略掉下面代码,自己独立实现
4  """
5
6  import requests
7  import re
8  import time
9  import random
10 from fake_useragent import UserAgent
11 import redis
12 from hashlib import md5
13 import sys
14 import pymysql
15
16 class TvSpider(object):
17     def __init__(self):
18         self.url = 'https://www.4567tv.tv/index.php/vod/show/id/5/page/{}.html'
19         self.r = redis.Redis(host='localhost', port=6379, db=0)
20         self.db = pymysql.connect('localhost', 'root', 'attack', 'tvdb', charset='utf8')
21         self.cursor = self.db.cursor()
22
23     def get_html(self, url):
24         headers = {'User-Agent': UserAgent().random}
25         html = requests.get(url=url, headers=headers).content.decode('utf-8')
26         return html

```



```

27
28     def regex_func(self, regex, html):
29         pattern = re.compile(regex, re.S)
30         r_list = pattern.findall(html)
31         return r_list
32
33     def parse_html(self, one_url):
34         one_html = self.get_html(one_url)
35         one_regex = '<li class="col-md-6 col-sm-4 col-xs-3">.*?<a class="stui-
vodlist__thumb lazyload" href="(.*?)".*?</li>'
36         href_list = self.regex_func(one_regex, one_html)
37         for href in href_list:
38             two_link = 'https://www.4567tv.tv' + href
39             finger = md5(two_link.encode()).hexdigest()
40             # sadd()添加成功返回 1 , 否则返回 0
41             result = self.r.sadd('tv:urls', finger)
42             if result:
43                 self.get_data(two_link)
44                 time.sleep(random.uniform(0, 1))
45             else:
46                 sys.exit('网站未更新数据')
47
48     def get_data(self, two_link):
49         two_html = self.get_html(two_link)
50         two_regex = '<div class="stui-content__detail">.*?<h1 class="title">(.*?)</h1>.*?
<span class="detail-content" style="display: none;">(.*?)</span>'
51         film_list = self.regex_func(two_regex, two_html)
52         if film_list:
53             film_name = film_list[0][0]
54             film_content = film_list[0][1]
55             ins = 'insert into tvtab values(%s,%s)'
56             self.cursor.execute(ins, [film_name, film_content])
57             self.db.commit()
58             print(film_name, film_content)
59
60
61     def run(self):
62         for i in range(1, 11):
63             one_url = self.url.format(i)
64             self.parse_html(one_url)
65
66
67 if __name__ == '__main__':
68     spider = TvSpider()
69     spider.run()

```

练习3 - 纵横中文网全站抓取

目标

- 1 1、纵横中文网 - 书库 - 全部作品
- 2 2、URL地址: <http://book.zongheng.com/store/c0/c0/b0/u0/p{/v9/s9/t0/u0/i1/ALL.html>

思路

- 1 1、一级页面: 提取 小说链接
- 2 2、二级页面: 提取 开始阅读对应的小说具体章节内容的链接
- 3 3、三级页面: 提取 目录 对应的链接 (链接中有此小说所有章节的明细及URL地址)
- 4 4、四级页面: 提取 此小说所有章节的链接
- 5 5、五级页面: 提取 具体的小说内容

准备工作

- 1 1、一级页面: 提取 小说链接
- 2 正则表达式: '

- 3 2、二级页面: 提取 开始阅读对应的小说具体章节内容的链接
- 4 正则表达式: '<div class="btn-group">.*?href="(.*?)".*?</div>'
- 5 3、三级页面: 提取 目录 对应的链接 (链接中有此小说所有章节的明细及URL地址)
- 6 目录正则表达式: '<div class="chap_btnbox">.*?目录'
- 7 名称正则表达式: '<body.*?bookName="(.*?)"'
- 8 4、四级页面: 提取 此小说所有章节的链接
- 9 正则表达式: '<li class=" col-4">.*?<a href="(.*?)".*?'
- 10 5、五级页面: 提取 具体的小说内容
- 11 正则表达式: '<div class="content".*?>(.*?)</div>'

代码实现

```
1 from urllib import request
2 import re
3 import time
4 import random
5
6 class NovelSpider(object):
7     def __init__(self):
8         # 主页的URL地址
9         self.url = 'http://book.zongheng.com/store/c0/c0/b0/u0/p{/v9/s9/t0/u0/i1/ALL.html'
10        self.headers = {
11            'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36'
12        }
13
14        # 功能函数1 - 获取html
15        def get_html(self, url):
16            req = request.Request(url=url, headers=self.headers)
17            res = request.urlopen(req)
18            html = res.read().decode()
19
20            return html
21
22        # 功能函数2 - xpath解析
23        def re_func(self, regex, html):
24            pattern = re.compile(regex, re.S)
25            r_list = pattern.findall(html)
26
```

```

27         return r_list
28
29     # 一级页面：提取小说链接
30     def parse_one_page(self, one_url):
31         one_html = self.get_html(url=one_url)
32         regex = '<div class="bookname">.*?href="(.*?)".*?</div>'
33         # one_link_list: [当页所有小说的链接]
34         one_link_list = self.re_func(regex, one_html)
35         for one_link in one_link_list:
36             # 将此小说的内容所有章节内容获取到
37             self.get_novel(one_link)
38
39     # 获取1个小说的所有章节内容
40     def get_novel(self, one_link):
41         two_html = self.get_html(url=one_link)
42         # 从开始阅读节点获取到小说具体内容的链接
43         regex = """"<div class="btn-group">.*?href="(.*?)".*?</div>""
44         two_link_list = self.re_func(regex, two_html)
45         two_link = two_link_list[0] if two_link_list else None
46         # 解析并提取此小说目录链接
47         if two_link:
48             self.get_novel_directory(two_link)
49
50     # 提取此小说目录链接
51     def get_novel_directory(self, two_link):
52         directory_html = self.get_html(url=two_link)
53         regex = '<div class="chap_btnbox">.*?<a href="(.*?)".*?>目录</a>'
54         directory_link_list = self.re_func(regex, directory_html)
55         directory_link = directory_link_list[0] if directory_link_list else None
56         # 获取小说名称
57         regex_name = '<body.*?bookName="(.*?)"'
58         name_list = self.re_func(regex_name, directory_html)
59         novel_name = name_list[0] if name_list else None
60         print(novel_name)
61         if directory_link and novel_name:
62             # 获取具体章节的目录链接
63             self.get_all_link(directory_link, novel_name)
64
65     # 获取具体章节的目录链接
66     def get_all_link(self, directory_link, novel_name):
67         directory_html = self.get_html(url=directory_link)
68         regex = '<li class=" col-4">.*?<a href="(.*?)".*?</a>'
69         novel_text_link_list = self.re_func(regex, directory_html)
70
71         for novel_text_link in novel_text_link_list:
72             # 获取具体小说章节内容
73             novel_text = self.get_novel_content(novel_text_link)
74             time.sleep(random.randint(1, 2))
75
76     # 获取具体小说章节内容
77     def get_novel_content(self, novel_text_link):
78         novel_text_html = self.get_html(url=novel_text_link)
79         regex = '<div class="content".*?(.*?)</div>'
80         novel_text = re.findall(regex, novel_text_html, re.S)
81         [0].replace('<p>', '').replace('</p>', '\n')
82         print(novel_text)

```

```
83         return novel_text
84
85
86     # 程序入口函数
87     def run(self):
88         for p in range(1,967):
89             url = self.url.format(p)
90             self.parse_one_page(url)
91
92 if __name__ == '__main__':
93     spider = NovelSpider()
94     spider.run()
```