练习1-电影天堂二级页面抓取

领取任务

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

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

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

■ 3、写正则表达式

■ 4、代码实现

```
from urllib import request
import re
from useragents import ua_list
import time
import random

class FilmSkySpider(object):
def __init__(self):
# 一级页面url地址
```

```
self.url = 'https://www.dvtt8.net/html/gndv/dvzz/list 23 {}.html'
10
11
12
      # 获取html功能函数
13
      def get_html(self,url):
       headers = {
14
15
          'User-Agent':random.choice(ua list)
16
17
       req = request.Request(url=url,headers=headers)
       res = request.urlopen(req)
18
       # 通过网站查看网页源码,查看网站charset='gb2312'
19
       # 如果遇到解码错误,识别不了一些字符,则 ignore 忽略掉
20
       html = res.read().decode('gb2312','ignore')
21
22
23
       return html
24
      # 正则解析功能函数
25
      def re func(self,re bds,html):
26
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'.*?'
       # one_page_list: ['/html/xxx','/html/xxx','/html/xxx']
36
       one page list = self.re func(re bds,html)
37
38
39
       for href in one page list:
40
         two url = 'https://www.dytt8.net' + href
         self.parse_two_page(two_url)
41
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 = {}
       html = self.get_html(two_url)
19
       re bds = r'<div class="title all"><h1><font color=#07519a>(.*?)</font></h1></div>.*?<td
50
    style="WORD-WRAP.*?>.*?>(.*?)</a>'
51
       # two_page_list: [('名称1','ftp://xxxx.mkv')]
52
       two page list = self.re func(re bds,html)
53
54
        item['name'] = two page list[0][0].strip()
55
       item['download'] = two_page_list[0][1].strip()
56
57
       print(item)
58
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
         time.sleep(random.uniform(1,3))
65
```

```
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

```
# 建库建表
create database filmskydb charset utf8;
use filmskydb;
create table request_finger(
finger char(32)
)charset=utf8;
create table filmtab(
name varchar(200),
download varchar(500)
)charset=utf8;
```

```
1 from urllib import request
2
    import re
   from useragents import ua list
   import time
    import random
   import pymysql
6
7
    from hashlib import md5
8
    import sys
9
10
   class FilmSkySpider(object):
     def init (self):
11
        # 一级页面url地址
12
        self.url = 'https://www.dytt8.net/html/gndy/dyzz/list_23_{}.html'
13
14
        self.db = pymysql.connect('localhost','root','attack','filmskydb',charset='utf8')
        self.cursor = self.db.cursor()
15
16
      # 获取html功能函数
17
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
        html = res.read().decode('gb2312','ignore')
26
27
        return html
28
29
```

```
30
      # 正则解析功能函数
      def re_func(self,re_bds,html):
31
32
        pattern = re.compile(re bds,re.S)
33
        r_list = pattern.findall(html)
34
35
        return r list
36
      # 获取数据函数 - html是一级页面响应内容
37
      def parse page(self, one url):
38
39
        html = self.get html(one url)
        re bds = r'.*?'
40
        # one page list: ['/html/xxx','/html/xxx','/html/xxx']
41
42
        one page list = self.re func(re bds,html)
43
44
        for href in one page list:
         two_url = 'https://www.dytt8.net' + href
45
          # 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
46
          sel = 'select finger from request finger where finger=%s'
47
48
          s = md5()
49
          s.update(two url.encode())
50
          finger = s.hexdigest()
51
          result = self.cursor.execute(sel,[finger])
         if not result:
52
53
           self.parse two page(two url)
           # uniform: 浮点数,爬取1个电影信息后sleep
54
55
           time.sleep(random.uniform(1, 3))
           ins = 'insert into request_finger values(%s)'
56
57
           self.cursor.execute(ins,[finger])
58
            self.db.commit()
59
          else:
60
            sys.exit('未更新')
61
62
63
      #解析二级页面数据
64
      def parse two page(self, two url):
        item = {}
65
66
        html = self.get html(two url)
        re_bds = r'<div class="title_all"><h1><font color=#07519a>(.*?)</font></h1></div>.*?<td
67
    style="WORD-WRAP.*?>.*?>(.*?)</a>'
        # two page list: [('名称1','ftp://xxxx.mkv')]
68
69
        two page list = self.re func(re bds,html)
70
71
        item['name'] = two_page_list[0][0].strip()
72
        item['download'] = two page list[0][1].strip()
        ins = 'insert into filmtab values(%s,%s)'
73
74
        film list = [
75
          item['name'],item['download']
76
        self.cursor.execute(ins,film_list)
77
78
        self.db.commit()
79
        print(film list)
80
81
82
      def run(self):
83
        for page in range(1,201):
84
         one_url = self.url.format(page)
85
          self.parse page(one url)
```

练习2 - 4567tv数据抓取

■ 领取任务

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

■ 实现步骤

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

■ 代码实现

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

```
import time
3
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
        def get_html(self,url):
11
12
            headers = { 'User-Agent':UserAgent().random }
            html = requests.get(url=url, headers=headers).content.decode('utf-8')
13
14
            return html
15
16
        def regex func(self,regex,html):
17
            pattern = re.compile(regex,re.S)
            r_list = pattern.findall(html)
18
19
            return r list
20
21
        def parse html(self,one url):
            one html = self.get html(one url)
22
23
            one regex = '.*?<a class="stui-
    vodlist thumb lazyload" href="(.*?)".*?'
            href list = self.regex func(one regex,one html)
24
25
            for href in href list:
                two link = 'https://www.4567tv.tv' + href
26
27
                self.get data(two link)
                time.sleep(random.uniform(0,1))
28
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]
            item['film_content'] = film_list[0][1]
36
37
            print(item)
38
39
        def run(self):
40
41
            for i in range(1,11):
42
                one_url = self.url.format(i)
43
                self.parse_html(one_url)
44
    if __name__ == '__main__':
45
46
        spider = TvSpider()
        spider.run()
47
```

■ 扩展 - 增量爬取

```
1 将数据存入MySQL数据库 - 增量爬取
2 # 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
```

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

■ 增量爬取 - MySQL

```
import requests
    import re
    import time
4
    import random
    from fake useragent import UserAgent
    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/{}.html'
            self.db = pymysql.connect('localhost', 'root', '123456', 'tvdb', charset='utf8')
13
            self.cursor = self.db.cursor()
14
15
        def get html(self, url):
16
            """功能函数1 - 获取相应内容"""
17
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)
            return r list
26
27
        def parse_html(self, one_url):
28
29
            """数据提取函数"""
30
            one_html = self.get_html(one_url)
            one_regex = '.*?<a class="stui-</pre>
31
    vodlist thumb lazyload" href="(.*?)".*?'
            href_list = self.regex_func(one_regex, one_html)
32
33
            for href in href list:
               two_link = 'https://www.4567tv.tv' + href
34
                # 对链接进行md5加密
35
               finger = md5(two link.encode()).hexdigest()
36
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):
            two html = self.get html(two link)
50
            two regex = '<div class="stui-content detail">.*?<h1 class="title">(.*?)</h1>.*?
51
    <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)'
            self.cursor.execute(ins, [film_name, film_content])
57
58
            self.db.commit()
            print(film name, film content)
59
60
        def run(self):
61
62
            for i in range(1, 11):
63
                one url = self.url.format(i)
64
                 self.parse html(one url)
65
    if __name__ == '__main__':
66
67
        spider = TvSpider()
        spider.run()
68
```

■ 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
   import requests
7
    import re
    import time
9
    import random
10
    from fake useragent import UserAgent
    import redis
11
12
    from hashlib import md5
13
    import sys
14
    import pymysql
15
16
    class TvSpider(object):
17
        def init (self):
            self.url = 'https://www.4567tv.tv/index.php/vod/show/id/5/page/{}.html'
18
            self.r = redis.Redis(host='localhost', port=6379, db=0)
19
            self.db = pymysql.connect('localhost','root','attack','tvdb',charset='utf8')
20
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)
            return r_list
31
32
33
        def parse_html(self, one_url):
34
            one html = self.get html(one url)
            one regex = '.*?<a class="stui-
35
    vodlist thumb lazyload" href="(.*?)".*?'
            href list = self.regex func(one regex, one html)
36
            for href in href list:
37
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
                result = self.r.sadd('tv:urls', finger)
41
42
                if result:
                    self.get_data(two_link)
43
44
                    time.sleep(random.uniform(0, 1))
45
                else:
                    sys.exit('网站未更新数据')
46
47
        def get data(self, two link):
48
49
            two html = self.get html(two link)
            two_regex = '<div class="stui-content__detail">.*?<h1 class="title">(.*?)</h1>.*?
50
    <span class="detail-content" style="display: none;">(.*?)</span>'
            film_list = self.regex_func(two_regex, two_html)
51
            if film list:
52
53
                film name = film list[0][0]
54
                film content = film list[0][1]
55
                ins = 'insert into tvtab values(%s,%s)'
                self.cursor.execute(ins, [film_name, film_content])
56
57
                self.db.commit()
                print(film_name, film_content)
58
59
60
61
        def run(self):
            for i in range(1, 11):
62
63
                one url = self.url.format(i)
                self.parse_html(one_url)
64
65
66
    if __name__ == '__main__':
67
68
        spider = TvSpider()
69
        spider.run()
```

练习3-纵横中文网全站抓取

```
1 、纵横中文网 - 书库 - 全部作品
2 以RL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 人四级页面: 提取 此小说所有章节的链接
5 无级页面: 提取 具体的小说内容
```

准备工作

```
1、一级页面: 提取 小说链接
   正则表达式: '<div class="bookname">.*?href="(.*?)".*?</div>'
   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
  |正则表达式: '<div class="btn-group">.*?href="(.*?)".*?</div>'
  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)
5
   目录正则表达式: '<div class="chap btnbox">.*?<a href="(.*?)".*?>目录</a>'
   名称正则表达式: '<body.*?bookName="(.*?)"'
7
  4、四级页面: 提取 此小说所有章节的链接
  正则表达式: '.*?<a href="(.*?)".*?</a>'
9
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
        # 功能函数1 - 获取html
14
15
        def get html(self,url):
            req = request.Request(url=url,headers=self.headers)
16
17
            res = request.urlopen(req)
            html = res.read().decode()
18
19
20
            return html
21
        # 功能函数2 - xpath解析
22
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):
            one_html = self.get_html(url=one_url)
31
            regex = '<div class="bookname">.*?href="(.*?)".*?</div>'
32
            # one_link_list: [当页所有小说的链接]
33
34
            one link list = self.re func(regex, one html)
35
            for one link in one link list:
                # 将此小说的内容所有章节内容获取到
36
37
                self.get novel(one link)
38
        # 获取1个小说的所有章节内容
39
        def get_novel(self,one_link):
40
41
            two html = self.get html(url=one link)
            # 从开始阅读节点获取到小说具体内容的链接
42
            regex = """<div class="btn-group">.*?href="(.*?)".*?</div>"""
43
            two_link_list = self.re_func(regex,two_html)
44
            two_link = two_link_list[0] if two_link_list else None
45
            # 解析并提取此小说目录链接
46
47
            if two_link:
48
                self.get novel directory(two link)
49
        # 提取此小说目录链接
50
        def get_novel_directory(self,two_link):
51
52
            directory_html = self.get_html(url=two_link)
            regex = '<div class="chap_btnbox">.*?<a href="(.*?)".*?>目录</a>'
53
54
            directory link list = self.re func(regex, directory html)
            directory_link = directory_link_list[0] if directory_link_list else None
55
            # 获取小说名称
56
57
            regex_name = '<body.*?bookName="(.*?)"'</pre>
            name_list = self.re_func(regex_name,directory_html)
58
59
            novel name = name list[0] if name list else None
60
            print(novel_name)
            if directory link and novel name:
61
                # 获取具体章节的目录链接
                self.get_all_link(directory_link,novel_name)
63
64
65
        # 获取具体章节的目录链接
        def get_all_link(self,directory_link,novel_name):
66
67
            directory_html = self.get_html(url=directory_link)
68
            regex = '.*?<a href="(.*?)".*?</a>'
            novel_text_link_list = self.re_func(regex,directory_html)
69
70
            for novel_text_link in novel_text_link_list:
71
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
79
            novel_text_html = self.get_html(url=novel_text_link)
80
            regex = '<div class="content".*?>(.*?)</div>'
81
            novel text = re.findall(regex,novel text html,re.S)
    [0].replace('','').replace('','\n')
82
            print(novel_text)
```

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