# **练习1 - 电影天堂二级页面抓取**

**领取任务**

# 地址  
电影天堂 - 2019年新片精品 - 更多  
# 目标  
电影名称、下载链接  
  
# 分析  
\*\*\*\*\*\*\*\*\*一级页面需抓取\*\*\*\*\*\*\*\*\*\*\*  
 1、电影详情页链接  
   
\*\*\*\*\*\*\*\*\*二级页面需抓取\*\*\*\*\*\*\*\*\*\*\*  
 1、电影名称  
 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**

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**

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

* **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.dytt8.net/html/gndy/dyzz/list\_23\_{}.html'  
  
 # 获取html功能函数  
 def get\_html(self,url):  
 headers = {  
 'User-Agent':random.choice(ua\_list)  
 }  
 req = request.Request(url=url,headers=headers)  
 res = request.urlopen(req)  
 # 通过网站查看网页源码,查看网站charset='gb2312'  
 # 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')  
  
 return html  
  
 # 正则解析功能函数  
 def re\_func(self,re\_bds,html):  
 pattern = re.compile(re\_bds,re.S)  
 r\_list = pattern.findall(html)  
  
 return r\_list  
  
 # 获取数据函数 - html是一级页面响应内容  
 def parse\_page(self,one\_url):  
 html = self.get\_html(one\_url)  
 re\_bds = r'<table width="100%".\*?<td width="5%".\*?<a href="(.\*?)".\*?ulink">.\*?</table>'  
 # one\_page\_list: ['/html/xxx','/html/xxx','/html/xxx']  
 one\_page\_list = self.re\_func(re\_bds,html)  
  
 for href in one\_page\_list:  
 two\_url = 'https://www.dytt8.net' + href  
 self.parse\_two\_page(two\_url)  
 # uniform: 浮点数,爬取1个电影信息后sleep  
 time.sleep(random.uniform(1, 3))  
  
  
 # 解析二级页面数据  
 def parse\_two\_page(self,two\_url):  
 item = {}  
 html = self.get\_html(two\_url)  
 re\_bds = r'<div class="title\_all"><h1><font color=#07519a>(.\*?)</font></h1></div>.\*?<td style="WORD-WRAP.\*?>.\*?>(.\*?)</a>'  
 # two\_page\_list: [('名称1','ftp://xxxx.mkv')]  
 two\_page\_list = self.re\_func(re\_bds,html)  
  
 item['name'] = two\_page\_list[0][0].strip()  
 item['download'] = two\_page\_list[0][1].strip()  
  
 print(item)  
  
  
 def main(self):  
 for page in range(1,201):  
 one\_url = self.url.format(page)  
 self.parse\_page(one\_url)  
 # uniform: 浮点数  
 time.sleep(random.uniform(1,3))  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 spider = FilmSkySpider()  
 spider.main()

* **5、练习**
* 把电影天堂数据存入MySQL数据库 - 增量爬取
* # 思路  
  # 1、MySQL中新建表 urltab,存储所有爬取过的链接的指纹  
  # 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;
* from urllib import request  
  import re  
  from useragents import ua\_list  
  import time  
  import random  
  import pymysql  
  from hashlib import md5  
  import sys  
    
  class FilmSkySpider(object):  
   def \_\_init\_\_(self):  
   # 一级页面url地址  
   self.url = 'https://www.dytt8.net/html/gndy/dyzz/list\_23\_{}.html'  
   self.db = pymysql.connect('localhost','root','attack','filmskydb',charset='utf8')  
   self.cursor = self.db.cursor()  
    
   # 获取html功能函数  
   def get\_html(self,url):  
   headers = {  
   'User-Agent':random.choice(ua\_list)  
   }  
   req = request.Request(url=url,headers=headers)  
   res = request.urlopen(req)  
   # 通过网站查看网页源码,查看网站charset='gb2312'  
   # 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')  
    
   return html  
    
   # 正则解析功能函数  
   def re\_func(self,re\_bds,html):  
   pattern = re.compile(re\_bds,re.S)  
   r\_list = pattern.findall(html)  
    
   return r\_list  
    
   # 获取数据函数 - html是一级页面响应内容  
   def parse\_page(self,one\_url):  
   html = self.get\_html(one\_url)  
   re\_bds = r'<table width="100%".\*?<td width="5%".\*?<a href="(.\*?)".\*?ulink">.\*?</table>'  
   # one\_page\_list: ['/html/xxx','/html/xxx','/html/xxx']  
   one\_page\_list = self.re\_func(re\_bds,html)  
    
   for href in one\_page\_list:  
   two\_url = 'https://www.dytt8.net' + href  
   # 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  
   sel = 'select finger from request\_finger where finger=%s'  
   s = md5()  
   s.update(two\_url.encode())  
   finger = s.hexdigest()  
   result = self.cursor.execute(sel,[finger])  
   if not result:  
   self.parse\_two\_page(two\_url)  
   # uniform: 浮点数,爬取1个电影信息后sleep  
   time.sleep(random.uniform(1, 3))  
   ins = 'insert into request\_finger values(%s)'  
   self.cursor.execute(ins,[finger])  
   self.db.commit()  
   else:  
   sys.exit('未更新')  
    
    
   # 解析二级页面数据  
   def parse\_two\_page(self,two\_url):  
   item = {}  
   html = self.get\_html(two\_url)  
   re\_bds = r'<div class="title\_all"><h1><font color=#07519a>(.\*?)</font></h1></div>.\*?<td style="WORD-WRAP.\*?>.\*?>(.\*?)</a>'  
   # two\_page\_list: [('名称1','ftp://xxxx.mkv')]  
   two\_page\_list = self.re\_func(re\_bds,html)  
    
   item['name'] = two\_page\_list[0][0].strip()  
   item['download'] = two\_page\_list[0][1].strip()  
   ins = 'insert into filmtab values(%s,%s)'  
   film\_list = [  
   item['name'],item['download']  
   ]  
   self.cursor.execute(ins,film\_list)  
   self.db.commit()  
   print(film\_list)  
    
    
   def run(self):  
   for page in range(1,201):  
   one\_url = self.url.format(page)  
   self.parse\_page(one\_url)  
   # uniform: 浮点数  
   time.sleep(random.uniform(1,3))  
    
  if \_\_name\_\_ == '\_\_main\_\_':  
   spider = FilmSkySpider()  
   spider.run()

# **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、爬取目标  
  电影名称、电影简介  
    
  # 3、爬取分析  
  \*\*\*\*\*\*\*\*\*一级页面需抓取\*\*\*\*\*\*\*\*\*\*\*  
  1、电影详情页的链接  
     
  \*\*\*\*\*\*\*\*\*二级页面需抓取\*\*\*\*\*\*\*\*\*\*\*  
  1、电影名称  
  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:  
  <li class="col-md-6 col-sm-4 col-xs-3">.\*?<a class="stui-vodlist\_\_thumb lazyload" href="(.\*?)".\*?</li>  
     
  二级页面正则:  
  <div class="stui-content\_\_detail">.\*?<h1 class="title">(.\*?)</h1>.\*?<span class="detail-content" style="display: none;">(.\*?)</span>  
    
  # 4. 代码实现
* **代码实现**
* import requests  
  import re  
  import time  
  import random  
  from fake\_useragent import UserAgent  
    
  class TvSpider(object):  
   def \_\_init\_\_(self):  
   self.url = 'https://www.4567tv.tv/index.php/vod/show/id/5/page/{}.html'  
    
   def get\_html(self,url):  
   headers = { 'User-Agent':UserAgent().random }  
   html = requests.get(url=url,headers=headers).content.decode('utf-8')  
   return html  
    
   def regex\_func(self,regex,html):  
   pattern = re.compile(regex,re.S)  
   r\_list = pattern.findall(html)  
   return r\_list  
    
   def parse\_html(self,one\_url):  
   one\_html = self.get\_html(one\_url)  
   one\_regex = '<li class="col-md-6 col-sm-4 col-xs-3">.\*?<a class="stui-vodlist\_\_thumb lazyload" href="(.\*?)".\*?</li>'  
   href\_list = self.regex\_func(one\_regex,one\_html)  
   for href in href\_list:  
   two\_link = 'https://www.4567tv.tv' + href  
   self.get\_data(two\_link)  
   time.sleep(random.uniform(0,1))  
    
   def get\_data(self,two\_link):  
   two\_html = self.get\_html(two\_link)  
   two\_regex = '<div class="stui-content\_\_detail">.\*?<h1 class="title">(.\*?)</h1>.\*?<span class="detail-content" style="display: none;">(.\*?)</span>'  
   film\_list = self.regex\_func(two\_regex,two\_html)  
   item = {}  
   item['film\_name'] = film\_list[0][0]  
   item['film\_content'] = film\_list[0][1]  
    
   print(item)  
    
   def run(self):  
   for i in range(1,11):  
   one\_url = self.url.format(i)  
   self.parse\_html(one\_url)  
    
  if \_\_name\_\_ == '\_\_main\_\_':  
   spider = TvSpider()  
   spider.run()
* **扩展 - 增量爬取**
* 将数据存入MySQL数据库 - 增量爬取  
    
  # 思路  
  1、MySQL中新建表 urltab,存储所有爬取过的链接的指纹  
  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;  
  use tvdb;  
  create table request\_finger(  
  finger char(32)  
  )charset=utf8;  
  create table tvtab(  
  name varchar(100),  
  comment varchar(1000)  
  )charset=utf8;
* **增量爬取 - MySQL**
* import requests  
  import re  
  import time  
  import random  
  from fake\_useragent import UserAgent  
  import pymysql  
  from hashlib import md5  
  import sys  
    
  class TvSpider(object):  
   def \_\_init\_\_(self):  
   self.url = 'https://www.4567tv.tv/index.php/vod/show/id/5/page/{}.html'  
   self.db = pymysql.connect('localhost', 'root', '123456', 'tvdb', charset='utf8')  
   self.cursor = self.db.cursor()  
    
   def get\_html(self, url):  
   """功能函数1 - 获取相应内容"""  
   headers = {'User-Agent': UserAgent().random}  
   html = requests.get(url=url, headers=headers).content.decode('utf-8')  
   return html  
    
   def regex\_func(self, regex, html):  
   """功能函数2 - 正则解析函数"""  
   pattern = re.compile(regex, re.S)  
   r\_list = pattern.findall(html)  
   return r\_list  
    
   def parse\_html(self, one\_url):  
   """数据提取函数"""  
   one\_html = self.get\_html(one\_url)  
   one\_regex = '<li class="col-md-6 col-sm-4 col-xs-3">.\*?<a class="stui-vodlist\_\_thumb lazyload" href="(.\*?)".\*?</li>'  
   href\_list = self.regex\_func(one\_regex, one\_html)  
   for href in href\_list:  
   two\_link = 'https://www.4567tv.tv' + href  
   # 对链接进行md5加密  
   finger = md5(two\_link.encode()).hexdigest()  
   sel = 'select finger from request\_finger where finger=%s'  
   result = self.cursor.execute(sel, [finger])  
   if not result:  
   self.get\_data(two\_link)  
   time.sleep(random.uniform(0, 1))  
   # 抓取完成后千万不要忘记存入指纹  
   ins = 'insert into request\_finger values(%s)'  
   self.cursor.execute(ins, [finger])  
   self.db.commit()  
   else:  
   sys.exit('网站未更新数据')  
    
   def get\_data(self, two\_link):  
   two\_html = self.get\_html(two\_link)  
   two\_regex = '<div class="stui-content\_\_detail">.\*?<h1 class="title">(.\*?)</h1>.\*?<span class="detail-content" style="display: none;">(.\*?)</span>'  
   film\_list = self.regex\_func(two\_regex, two\_html)  
    
   film\_name = film\_list[0][0]  
   film\_content = film\_list[0][1]  
   ins = 'insert into tvtab values(%s,%s)'  
   self.cursor.execute(ins, [film\_name, film\_content])  
   self.db.commit()  
   print(film\_name, film\_content)  
    
   def run(self):  
   for i in range(1, 11):  
   one\_url = self.url.format(i)  
   self.parse\_html(one\_url)  
    
  if \_\_name\_\_ == '\_\_main\_\_':  
   spider = TvSpider()  
   spider.run()
* **能不能使用redis来实现增量**
* """  
   提示: 使用redis中的集合,sadd()方法,添加成功返回1,否则返回0  
   请各位大佬忽略掉下面代码,自己独立实现  
  """  
    
  import requests  
  import re  
  import time  
  import random  
  from fake\_useragent import UserAgent  
  import redis  
  from hashlib import md5  
  import sys  
  import pymysql  
    
  class TvSpider(object):  
   def \_\_init\_\_(self):  
   self.url = 'https://www.4567tv.tv/index.php/vod/show/id/5/page/{}.html'  
   self.r = redis.Redis(host='localhost', port=6379, db=0)  
   self.db = pymysql.connect('localhost','root','attack','tvdb',charset='utf8')  
   self.cursor = self.db.cursor()  
    
   def get\_html(self, url):  
   headers = {'User-Agent': UserAgent().random}  
   html = requests.get(url=url, headers=headers).content.decode('utf-8')  
   return html  
    
   def regex\_func(self, regex, html):  
   pattern = re.compile(regex, re.S)  
   r\_list = pattern.findall(html)  
   return r\_list  
    
   def parse\_html(self, one\_url):  
   one\_html = self.get\_html(one\_url)  
   one\_regex = '<li class="col-md-6 col-sm-4 col-xs-3">.\*?<a class="stui-vodlist\_\_thumb lazyload" href="(.\*?)".\*?</li>'  
   href\_list = self.regex\_func(one\_regex, one\_html)  
   for href in href\_list:  
   two\_link = 'https://www.4567tv.tv' + href  
   finger = md5(two\_link.encode()).hexdigest()  
   # sadd()添加成功返回 1 , 否则返回 0  
   result = self.r.sadd('tv:urls', finger)  
   if result:  
   self.get\_data(two\_link)  
   time.sleep(random.uniform(0, 1))  
   else:  
   sys.exit('网站未更新数据')  
    
   def get\_data(self, two\_link):  
   two\_html = self.get\_html(two\_link)  
   two\_regex = '<div class="stui-content\_\_detail">.\*?<h1 class="title">(.\*?)</h1>.\*?<span class="detail-content" style="display: none;">(.\*?)</span>'  
   film\_list = self.regex\_func(two\_regex, two\_html)  
   if film\_list:  
   film\_name = film\_list[0][0]  
   film\_content = film\_list[0][1]  
   ins = 'insert into tvtab values(%s,%s)'  
   self.cursor.execute(ins, [film\_name, film\_content])  
   self.db.commit()  
   print(film\_name, film\_content)  
    
    
   def run(self):  
   for i in range(1, 11):  
   one\_url = self.url.format(i)  
   self.parse\_html(one\_url)  
    
    
  if \_\_name\_\_ == '\_\_main\_\_':  
   spider = TvSpider()  
   spider.run()

# **练习3 - 纵横中文网全站抓取**

**目标**

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

**思路**

1、一级页面：提取 小说链接  
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、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、四级页面：提取 此小说所有章节的链接  
5、五级页面：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  
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）  
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="(.\*?)"'  
4、四级页面：提取 此小说所有章节的链接  
正则表达式：'<li class=" col-4">.\*?<a href="(.\*?)".\*?</a>'  
5、五级页面：提取 具体的小说内容  
正则表达式：'<div class="content".\*?>(.\*?)</div>'

**代码实现**

from urllib import request  
import re  
import time  
import random  
  
class NovelSpider(object):  
 def \_\_init\_\_(self):  
 # 主页的URL地址  
 self.url = 'http://book.zongheng.com/store/c0/c0/b0/u0/p{}/v9/s9/t0/u0/i1/ALL.html'  
 self.headers = {  
 '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'  
 }  
  
 # 功能函数1 - 获取html  
 def get\_html(self,url):  
 req = request.Request(url=url,headers=self.headers)  
 res = request.urlopen(req)  
 html = res.read().decode()  
  
 return html  
  
 # 功能函数2 - xpath解析  
 def re\_func(self,regex,html):  
 pattern = re.compile(regex,re.S)  
 r\_list = pattern.findall(html)  
  
 return r\_list  
  
 # 一级页面：提取小说链接  
 def parse\_one\_page(self,one\_url):  
 one\_html = self.get\_html(url=one\_url)  
 regex = '<div class="bookname">.\*?href="(.\*?)".\*?</div>'  
 # one\_link\_list: [当页所有小说的链接]  
 one\_link\_list = self.re\_func(regex,one\_html)  
 for one\_link in one\_link\_list:  
 # 将此小说的内容所有章节内容获取到  
 self.get\_novel(one\_link)  
  
 # 获取1个小说的所有章节内容  
 def get\_novel(self,one\_link):  
 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  
 regex = """<div class="btn-group">.\*?href="(.\*?)".\*?</div>"""  
 two\_link\_list = self.re\_func(regex,two\_html)  
 two\_link = two\_link\_list[0] if two\_link\_list else None  
 # 解析并提取此小说目录链接  
 if two\_link:  
 self.get\_novel\_directory(two\_link)  
  
 # 提取此小说目录链接  
 def get\_novel\_directory(self,two\_link):  
 directory\_html = self.get\_html(url=two\_link)  
 regex = '<div class="chap\_btnbox">.\*?<a href="(.\*?)".\*?>目录</a>'  
 directory\_link\_list = self.re\_func(regex,directory\_html)  
 directory\_link = directory\_link\_list[0] if directory\_link\_list else None  
 # 获取小说名称  
 regex\_name = '<body.\*?bookName="(.\*?)"'  
 name\_list = self.re\_func(regex\_name,directory\_html)  
 novel\_name = name\_list[0] if name\_list else None  
 print(novel\_name)  
 if directory\_link and novel\_name:  
 # 获取具体章节的目录链接  
 self.get\_all\_link(directory\_link,novel\_name)  
  
 # 获取具体章节的目录链接  
 def get\_all\_link(self,directory\_link,novel\_name):  
 directory\_html = self.get\_html(url=directory\_link)  
 regex = '<li class=" col-4">.\*?<a href="(.\*?)".\*?</a>'  
 novel\_text\_link\_list = self.re\_func(regex,directory\_html)  
  
 for novel\_text\_link in novel\_text\_link\_list:  
 # 获取具体小说章节内容  
 novel\_text = self.get\_novel\_content(novel\_text\_link)  
 time.sleep(random.randint(1,2))  
  
  
 # 获取具体小说章节内容  
 def get\_novel\_content(self,novel\_text\_link):  
 novel\_text\_html = self.get\_html(url=novel\_text\_link)  
 regex = '<div class="content".\*?>(.\*?)</div>'  
 novel\_text = re.findall(regex,novel\_text\_html,re.S)[0].replace('<p>','').replace('</p>','\n')  
 print(novel\_text)  
 return novel\_text  
  
  
 # 程序入口函数  
 def run(self):  
 for p in range(1,967):  
 url = self.url.format(p)  
 self.parse\_one\_page(url)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 spider = NovelSpider()  
 spider.run()