

爬虫期中作业

爬虫程序

豆瓣电影top

```
from lxml import etree
import requests
import MySQLdb

def get_movies(i):
    #设置请求头
    headers={
        'user-agent': 'Mozilla/5.0 (Windows NT 6.1;win64;x64) AppleWebKit/537.36 (KHTML,like Gecko) Chrome/52.0.2743.82 Safari/537.36',
        'Host': 'movie.douban.com'
    }
    movie_list=[]
    for a in range(i):
        link='https://movie.douban.com/top250?start='+ str(a*25)
        r=requests.get(link,headers=headers)
        s = etree.HTML(r.text)
        All_movie = s.xpath('//*[id="content"]/div/div[1]/ol/li')
        for Movie in All_movie:
            conn=MySQLdb.connect(host='localhost',
                                user='dxj',
                                passwd='a81877568',
                                db='oa',
                                charset='utf8')

            cur=conn.cursor()
            M_name=Movie.xpath('./div/div[2]/div[1]/a/span[1]/text()')
            name=", ".join(map(str, M_name))
            M_star=Movie.xpath('./div/div[2]/div[2]/div/span[2]/text()')
            star=", ".join(map(str, M_star))
            M_href=Movie.xpath('./div/div[2]/div[1]/a/@href')
            href=", ".join(map(str, M_href))
            #将数据插入相应的电影表中
            sql="INSERT INTO top_movie(m_name,m_star,m_href) VALUES ('%s','%s','%s')"%(name,star,href)
            movie_list.append(M_name)
            cur.execute(sql)
            cur.close()
            conn.commit()
            conn.close()

    #返回电影列表
    return movie_list
if __name__ == "__main__":
    movies=get_movies(4)
    print (movies)
```

所遇到问题:

1. xpath返回的内容为list

所以我尝试用代码将其转为字符串, 如:

```
M_name=Movie.xpath('./div/div[2]/div[1]/a/span[1]/text()')
name=", ".join(map(str, M_name))
```

2.

京东手机

```

import requests
from lxml import etree
import time
import MySQLdb
#定义函数抓取每页前60条商品信息
def get_mobile(n):
    #构造每一页的url变化
    url='https://search.jd.com/Search?keyword=%E6%89%8B%E6%9C%BA&enc=utf-8&qrst=1&rt=1&stop=1&vt=2&cid2=653&cid3=655&page='+str(2*n-1)
    headers = {'authority': 'search.jd.com',
                'method': 'GET',
                'scheme': 'https',
                'referer': 'https://search.jd.com/Search?keyword=%E6%89%8B%E6%9C%BA&enc=utf-8&qrst=1&rt=1&stop=1&vt=2&wq=%E6%89%8B%E6%9C%BA&cid2=653&cid3=655&page=1&s=58&click=0',
                'user-agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.119 Safari/537.36',
                'x-requested-with': 'XMLHttpRequest',
                'Cookie': 'qrsc=3; pinId=RAGa4xMovrs; xtest=1210.cf6b6759; ipLocation=%u5E7F%u4E1C; _jrda=5; TrackID=1aUdbC9HHS2MdEzabUYeED1iDJaLwwBAfGBfyIHJZCLWkfwaB_KHKIMX9Vj9_2wUakxuSLA09AftB2U0SsAD-mXIh5rIfuDiShSNhZcsJvg; shshshfpa=17943c91-d534-104f-a035-6e1719740bb6-1525571955; shshshfpb=2f200f7c5265e4af999b95b20d90e6618559f7251020a80ea1aee61500; cn=0; 3AB9D23F7A4B3C9B=QFOFIDQSiC7TZDQ7U4RPNYNFQN7S26SFCQGTc3YU5UZQJZUBNPEXMX703R7SIRBTTJ72AXC4S3IJ46ESBLTNHD37U; ipLoc-djd=19-1607-3638-3638.608841570; __jdu=930036140; user-key=31a7628c-a9b2-44b0-8147-f10a9e597d6f; areaId=19; __jdv=122270672|direct|-|none|-|1529893590075; PCSYCityID=25; mt_xid=v2_52007VwsQ1LxavVoaSCLUA2YLEAdbwk5YSk9MQAA0BBZOVQ0ADWNLGLUAZwQXVQpaAlkvShhCDHsCFU5eXENaGkIZWg5NayJqbVhiwR9BG1UNZwowY1ldVF0%3D; __jdc=122270672; shshshfp=72ec41b59960ea9a26956307465948f6; rkv=v0700; __jda=122270672.930036140.-.1529979524.1529984840.85; __jdb=122270672.1.930036140|85.1529984840; shshshsID=f797fbad20f4e576e9c30d1c381ecbb1_1_1529984840145'}

    r = requests.get(url, headers=headers)
    #指定编码方式, 不然会出现乱码
    r.encoding='utf-8'
    htmltext = etree.HTML(r.text)
    #定位到每一个商品标签li
    datas=htmltext.xpath('//li[contains(@class,"gl-item")]')
    #将数据存储到mysql数据库
    mobile_list=[]
    for data in datas:
        conn=MySQLdb.connect(host='localhost',
                              user='dxj',
                              passwd='a81877568',
                              db='python',
                              charset='utf8')

        cur=conn.cursor()
        array_price = data.xpath('./div/div[3]/strong/i/text()[1]')
        p_price = ",".join(map(str, array_price))
        array_comment = data.xpath('./div/div[4]/a/i/text()')
        p_comment = ",".join(map(str, array_comment))
        array_name = data.xpath('./div/div[4]/a/em/text()')
        p_name = ",".join(map(str, array_name))
        mobile_list.append(p_name)
        sql="INSERT INTO jd_mobile(m_name,m_price,m_comment) VALUES('%s','%s','%s')"%
        (p_name,p_price,p_comment)
        cur.execute(sql)
        cur.close()
        conn.commit()
        conn.close()
    return mobile_list
#定义函数抓取每页后30条商品信息

def get_last(n):
    #获取当前的Unix时间戳, 并且保留小数点后5位
    a=time.time()
    b='%5f'%a
    url='https://search.jd.com/s_new.php?keyword=%E6%89%8B%E6%9C%BA&enc=utf-8&qrst=1&rt=1&stop=1&vt=2&wq=%E6%89%8B%E6%9C%BA&cid2=653&cid3=655&page='+str(2*n)+'&s='+str(48*n-20)+'&scrolling=y&log_id='+str(b)
    head={'authority': 'search.jd.com',
          'method': 'GET',

```

```

'path': '/s_new.php?keyword=%E6%89%8B%E6%9C%BA&enc=utf-
8&qrst=1&rt=1&stop=1&vt=2&wq=%E6%89%8B%E6%9C%BA&cid2=653&cid3=655&page=4&s=84&scrolling=y&log_id=1529828
108.22071&tpl=3_M&show_items=7651927,7367120,7056868,7419252,6001239,5934182,4554969,3893501,7421462,657
7495,26480543553,7345757,4483120,6176077,6932795,7336429,5963066,5283387,25722468892,7425622,4768461',
'scheme': 'https',
'referer': 'https://search.jd.com/Search?keyword=%E6%89%8B%E6%9C%BA&enc=utf-
8&qrst=1&rt=1&stop=1&vt=2&wq=%E6%89%8B%E6%9C%BA&cid2=653&cid3=655&page=3&s=58&click=0',
'user-agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/72.0.3626.119 Safari/537.36',
'x-requested-with': 'XMLHttpRequest',
'Cookie': 'qrsc=3; pinId=RAGa4xM0vrs; xtest=1210.cf6b6759; ipLocation=%u5E7F%u4E1C; _jrda=5;
TrackID=1auDbc9HHS2MdEzabuyEyED1iDJaLWWBAFGBfyIHJZCLWkfWab_KHKIMX9Vj9_2wUakxuSLAO9AftB2U0SsAD-
mXiH5rIfuDiSHSNhZcsJvg; shshshfpa=17943c91-d534-104f-a035-6e1719740bb6-1525571955;
shshshfpb=2f200f7c5265e4af999b95b20d90e6618559f7251020a80eaaee61500; cn=0;
3AB9D23F7A4B3C9B=QFOFIDQSiC7TZDQ7U4RPNYNFQN7S26SFCQGTC3YU5UZQJZUBNPEXMX7O3R7SIRBTTJ72AXC453IJ46ESBLTNHD
37U; ipLoc-djd=19-1607-3638-3638.608841570; __jdu=930036140; user-key=31a7628c-a9b2-44b0-8147-
f10a9e597d6f; areaId=19; __jdv=122270672|direct|-|none|-|1529893590075; PCSYCityID=25;
mt_xid=V2_52007VwsQU1xavVoaSC1UA2YLEAdbwk5YSk9MQAA0BBZOVQ0ADWNLG1UAZWQXVQpaA1kvShhcDHSCFU5eXENaGkIZwg5Na
yJQbvhiWR9BG1UNZwowy1ldvF0%3D; __jdc=122270672; shshshfp=72ec41b59960ea9a26956307465948f6; rkv=v0700;
__jda=122270672.930036140.-.1529979524.1529984840.85; __jdb=122270672.1.930036140|85.1529984840;
shshshsID=f797fbad20f4e576e9c30d1c381ecbb1_1_1529984840145'
}
r = requests.get(url, headers=head)
r.encoding = 'utf-8'
html1 = etree.HTML(r.text)
datas = html1.xpath('//li[contains(@class,"gl-item")]')
#将数据存储到mysql数据库
mobile_list2=[]
for data in datas:
    conn=MySQLdb.connect(host='localhost',
                           user='dxj',
                           passwd='a81877568',
                           db='python',
                           charset='utf8')

    cur=conn.cursor()
    array_price =data.xpath('./div/div[3]/strong/i/text()[1]')
    p_price = ",".join(map(str, array_price))
    array_comment =data.xpath('./div/div[4]/a/i/text()')
    p_comment = ",".join(map(str, array_comment))
    array_name = data.xpath('./div/div[4]/a/em/text()')
    p_name = ",".join(map(str, array_name))
    mobile_list2.append(p_name)
    sql="INSERT INTO jd_mobile(m_name,m_price,m_comment) VALUES('%s','%s','%s')"%
(p_name,p_price,p_comment)
    cur.execute(sql)
    cur.close()
    conn.commit()
    conn.close()
return mobile_list2

if __name__=='__main__':
    for i in range(1,2):
        #下面的print函数主要是为了方便查看当前抓到第几页了
        print('*****')
        try:
            print(' First_Page: ' + str(i))
            mobile_list1=get_mobile(i)
            mobile_list2=get_last(i)
            print(' Finish')
            print(mobile_list1)
            print(mobile_list2)
        except Exception as e:
            print(e)
        print('-----')

```

部分借鉴了网上的内容

前三十条是直接显示，所以可以直接爬取，后三十为动态加载通过找到对应的API接口，然后构造请求来模拟浏览器。

Django

搭建环境

进入venv环境

创建对应Django文件夹，在对应文件夹下输入命令


1. 创建虚拟环境

```
python -m venv venv
```

2. 运行文件

```
venv/Scripts/activate.bat
```

进入如下图所示界面

 命令提示符

```
(venv) C:\Users\asus\Desktop\Django\venv\Scripts>cd. .  
  
(venv) C:\Users\asus\Desktop\Django\venv>cd. .  
  
(venv) C:\Users\asus\Desktop\Django>
```

安装相应库

1. 更新包管理工具pip

```
(venv) C:\Users\asus\Desktop\Django>python -m pip install --upgrade pip
```

2. 下载Django与PyMysql

```
pip install django==2.0  
pip install PyMysql
```

下载的版本是Django2.0与PyMysql0.9.3

可通过pip list查看对应版本

构建项目与管理

创建项目

```
django-admin startproject Pyspider
```

创建动态页面

```
python manage.py startapp Spider
```

连接mysql数据库

打开settings文件

```
#将Spider添加已安装的项目中  
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',
```

```
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'Spider',
]
'''省略'''
#修改连接内容
DATABASES = {
'default': {
'ENGINE': 'django.db.backends.mysql',
#已创建好的数据库
'NAME': 'oa',
'HOST': 'localhost',
'PORT': 3306,
'USER': 'dxj',
'PASSWORD': 'a81877568',
}
}
```

修改项目的__init__.py文件并加入如下所示的代码，这段代码的作用是将PyMySQL视为MySQLdb来使用，从而避免Django找不到连接MySQL的客户端工具而询问你：“Did you install mysqlclient?”

```
import pymysql

pymysql.install_as_MySQLdb()
```

尝试使用python manage.py migrate实现数据库迁移，为应用程序创建对应的数据表

发生报错

```
django.db.utils.ProgrammingError: (1064, "You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ';' SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED' at line 1")
```

查询资料发现可能与我mysql版本有关

在DATABASES中添加了一句'OPTIONS':{'isolation_level':None} 运行成功

```
(venv) C:\Users\asus\Desktop\Django\Pyspider>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK
```

初次运行

```
python manage.py runserver
#运行后进入http://127.0.0.1:8000/
```



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

后台管理模型

1. 建立超级用户

```
python manage.py createsuperuser
```

```
(venv) C:\Users\asus\Desktop\Django\Pyspider>python manage.py createsuperuser
Username (leave blank to use 'asus'): dxj
Email address: 1054242248@qq.com
Password:
Password (again):
Superuser created successfully.
```

观察数据库发现用户创建成功并且密码加密了

2. 登录后台管理系统

样式原先无法显示

请登录

[忘记了您的密码或用户名？](#)

查询大量资料，尝试数遍后发现修改base.html的标签就可以正常显示

Django 管理

欢迎, **DXJ** | [查看站点](#) / [修改密码](#) / [注销](#)

站点管理

认证和授权

用户

[+ 增加](#) [✎ 修改](#)

组

[+ 增加](#) [✎ 修改](#)

最近动作

我的动作

无可用的

3. 创建models

先尝试豆瓣电影top250

编写models文件

```

from django.db import models

# Create your models here.
class Movie(models.Model):
    #列的内容
    Movie_name = models.CharField(max_length=255, db_column='m_name', verbose_name='电影名称')
    class Meta:
        db_table = 'top_movie'

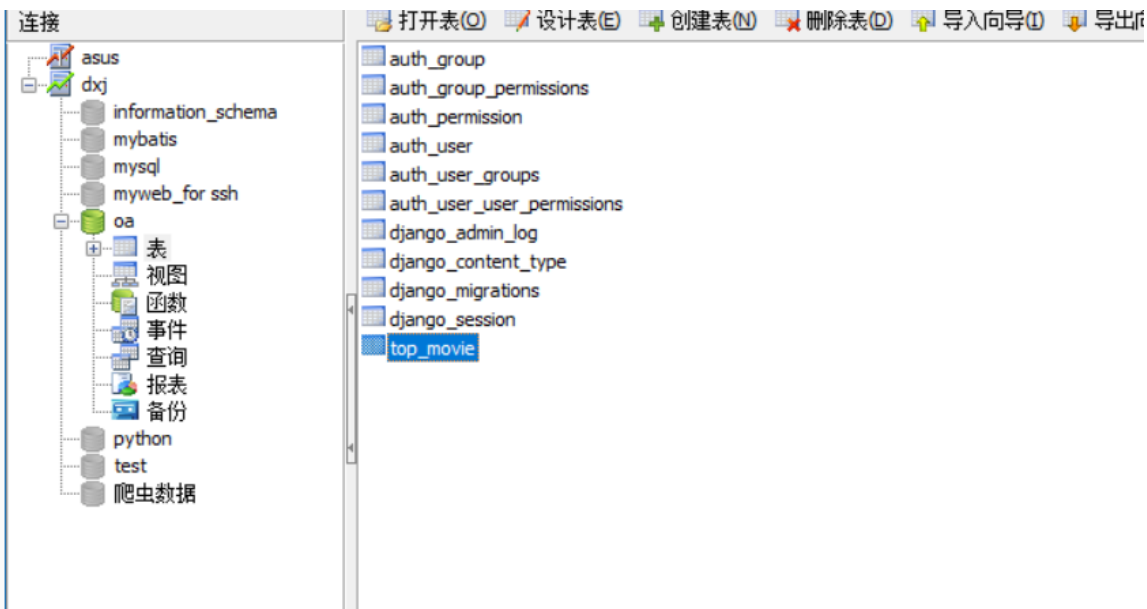
```

通过模型创建数据表。

```

python manage.py makemigrations Spider
python manage.py migrate

```



运行完毕发现创建成功

注册模型类

```

from django.contrib import admin

from Spider.models import Movie

admin.site.register(Movie)

```

站点管理

SPIDER		
Movies	+ 增加	✎ 修改
认证和授权		
用户	+ 增加	✎ 修改
组	+ 增加	✎ 修改

最近动作

我的动作

无可用的

进行测试

增加 movie

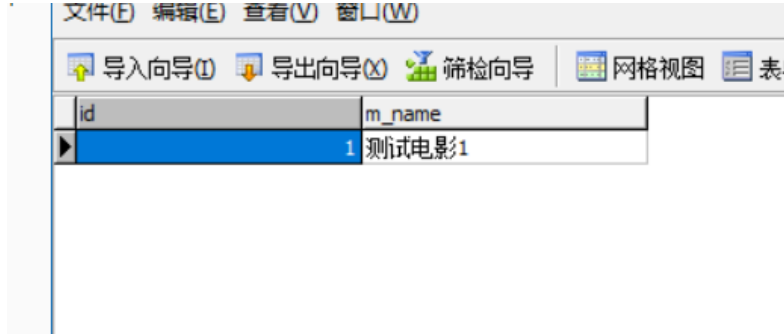
电影名称:

测试电影1

保存并增加另一个

保存并继续编辑

保存



管理界面优化

```
from django.contrib import admin

from spider.models import Movie

class MovieAdmin(admin.ModelAdmin):
    list_display = ('id', 'Movie_name')
    search_fields = ('id', 'Movie_name')

admin.site.register(Movie, MovieAdmin)
```

Q 搜索

动作 执行 3 个中 0 个被选

<input type="checkbox"/>	ID	电影名称
<input type="checkbox"/>	3	测试电影3
<input type="checkbox"/>	2	测试电影2
<input type="checkbox"/>	1	测试电影1

3 movies

页面设计

修改views.py

以豆瓣为例

```
#coding:utf-8
from django.shortcuts import render
from spider.models import Movie
# Create your views here.
def show_movie(request):
    ctx = {'movie_list': Movie.objects.all()}
    return render(request, 'demo/movie.html', ctx)
```

修改urls.py

为模板页增加路径


```
from django.contrib import admin
from django.urls import path

from Spider import views
urlpatterns = [

    path('admin/', admin.site.urls),
    path('movie/', views.show_movie),
]
```

设计模板页

豆瓣电影

{% for movie in movie_list %}

{{movie.Movie_name }}

{{movie.Movie_star}}

{{movie.Movie_href}}

{% endfor %}

结果展示

豆瓣电影

肖申克的救赎

9.6

<https://movie.douban.com/subject/1292052/>

霸王别姬

9.6

<https://movie.douban.com/subject/1291546/>

这个杀手不太冷

9.4

<https://movie.douban.com/subject/1295644/>

阿甘正传

9.4

<https://movie.douban.com/subject/1292720/>

美丽人生

最终结果

豆瓣

结果之前已经差不多展示了就不再展示

京东

京东手机内容爬取与豆瓣大体类似

在已有环境下流程



结果如下

京东手机

手机名称介绍: vivoZ3 6GB+64GB大内存 领券立减100元

促销活动内容:
手机价格: 1598.00

手机名称介绍: 荣耀8X 千元屏霸 91%屏占比 2000万AI双摄 4GB+64GB 幻夜黑 移动联通电信4G全面屏, 双卡双待

促销活动内容: 限时优惠1299! 麒麟710处理器, 2000万AI双摄, 人脸+指纹双识别! 荣耀爆品特惠, 选品质, 购荣耀~
手机价格: 1298.00

手机名称介绍: Apple iPhone XR (A2108) 128GB 黑色 移动联通电信4G, 双卡双待

促销活动内容: 【五一假期提前放价, 抢券立减400元】小屏经典iPhone7仅售3199元, iPhoneXR低至4799元! 白条12期免息券27日截止, 限时限量速抢!
手机价格: 5699.00

总结感想

在做的途中可以说大小bug不断, 各种版本不兼容问题等等, 所幸的是最终还是完成了。虽然说最后的展示并不好看, 也没有做什么花里胡哨的东西, 但是总的过程还是大致上了解了。毕竟这还只是初步的内容, 后期会逐步完善, 如:

1. 爬取图片进行保存
2. 将爬虫程序写入Django框架中, 而不是独立出去
3. 为每条内容添加链接, 转入详细介绍的页面
4. 融入Ajax内容对电影手机进行评价
5. 实现“用户注册”和“用户登录”的功能。
6. 尝试使用多线程爬取网页

还算任重道远, 仍需努力, 不过目前就到此为止吧~

OVER~