



# 温州大学瓯江学院

WENZHOU UNIVERSITY OUJIANG COLLEGE

## 《爬虫期中作业》

题    目：\_\_\_\_\_网络爬虫\_\_\_\_\_

分    院：\_\_\_\_\_数学与信息工程学院\_\_\_\_\_

班    级：\_\_\_\_\_16 计算机科学与技术二班\_\_\_\_\_

姓    名：\_\_\_\_\_邱朝海\_\_\_\_\_

学    号：\_\_\_\_\_16219111201\_\_\_\_\_

完成日期：\_\_\_\_\_2019 年 4 月 26 日\_\_\_\_\_

温州大学瓯江学院教务部

二〇一二年十一月制

# 目录

目录.....	2
一、 实验要求介绍 .....	3
1、实验要求: .....	3
2、开发工具与编程语言: .....	3
二、 设计过程简单介绍.....	4
1、静态爬虫: (只提供部分代码) .....	4
2、动态爬虫: (只提供部分代码) .....	4
3、数据库: (数据库中的两表都是事先创建好的) .....	7
4、Django: (我将静态和动态的代码均放在同一个框架内, 所以文件命名可以无视) .....	8
三、 运行截图 .....	13
1、运行前的注意事项: .....	13
2、截图: .....	14
四、 未解决的问题 .....	17

# 一、实验要求介绍

## 1、实验要求：

(1) 模仿书本 P36 “Request 爬虫实践：TOP250 电影数据”爬取相关数据，并数入据存入数据库中。(数据库软件自选)

(2) 建立 Django 项目：

基本要求：可运行 Django 项目；

进阶要求：展示爬虫信息的 Django 项目

高阶要求：基于 BeautifulSoup 或其他网页模拟的爬虫的 Django 项目

(3) 使用 selenium 爬取动态网站—京东手机。

(4) 上传代码到 Github

## 2、开发工具与编程语言：

这里只是简单介绍：

使用软件：Visual Studio code , MySQL



Python 语言中的相关模块和类库：

**Request:** Request 支持 HTTP 连接保持和连接池，支持使用 cookie 保持会话，支持文件上传，支持自动响应内容的编码，支持国际化的 URL 和 POST 数据自动编码。在 python 内置模块的基础上进行了高度的封装，从而使得 python 进行网络请求时，变得人性化，使用 Requests 可以轻而易举的完成浏览器可有的任何操作。

**BeautifulSoup:** BeautifulSoup 提供一些简单的、python 式的函数用来处理导航、搜索、修改分析树等功能。它是一个工具箱，通过解析文档为用户提供需要抓取的数据，因为简单，所以不需要多少代码就可以写出一个完整的应用程序。简单来说，Beautiful Soup 是 python 的一个库，最主要的功能是从网页抓取数据。

**Pymysql:** 在 python3.x 中，可以使用 pymysql 来 MySQL 数据库的连接，并实现数据库的各种操作。

**Selenium:** selenium 是一套完整 web 应用程序测试系统, 包含了测试的录制 (selenium IDE), 编写及运行 (Selenium Remote Control) 和测试的并行处理 (Selenium Grid)。Selenium 的核心 Selenium Core 基于 JsUnit, 完全由 JavaScript 编写, 因此可以用于任何支持 JavaScript 的浏览器上。selenium 可以模拟真实浏览器, 自动化测试工具, 支持多种浏览器, 爬虫中主要用来解决 JavaScript 渲染问题。

其余的还有 **time**, **os** 两个类库就不介绍。

## 二、设计过程简单介绍

### 1、静态爬虫: (只提供部分代码)

用 Request 读取网页源代码:

```
def getHTMLText(url):
    try:
        res=requests.get(url,timeout=60)
        res.raise_for_status()
        res.encoding=res.apparent_encoding
        return res.text
    except:
        return "产生异常!"
```

用 BeautifulSoup 分析源代码, 爬取需要的数据:

```
def fillList(ulist,html):
    soup=BeautifulSoup(html,"html.parser")
    movies=soup.find('ol',class_="grid_view")
    if movies:
        for mv in movies.find_all('li'):
            href=mv.find('div',class_="hd").a.attrs['href']
            title=mv.find('div',class_="hd").a.span.get_text()
            quote=mv.find('p',class_="quote").span.get_text()
            ulist.append([href,title,quote])
        return ulist
    else:
        print("找不到参数!")
```

设计相关代码, 传入数据库 (Mysql):

```
def inputMysql(ulist):    #存入数据库
    conn=pymysql.connect(host='localhost',user='root',passwd='admin',db='douban',charset="utf8",cursorclass=pymysql.cursors.DictCursor)
    cur=conn.cursor()
    try:
        for i in ulist:
            cur.execute("insert into movies (链接,标题,简介) values (%s,%s,%s)",(i[0],i[1],i[2]))
        cur.close()
        conn.commit()
        conn.close()
        print("数据已传入数据库!")
    except:
        print("数据传输失败!")
```

### 2、动态爬虫: (只提供部分代码)

因为用的是 Chrome 浏览器, 下载对应版本的插件, 这里因为找不到当初装 Python

的安装目录，所以只好自己在代码中指定路径：

```
url="https://www.jd.com"
phone=JDphone()
phone.Spider(url,"手机",tlist)
```

```
class JDphone():
    def Open_web(self,url,key):    #找到查询板块，模拟浏览器输入“手机”搜索手机页面
        chromedriver="D:/平时文件/爬虫/京东手机/chromedriver.exe"
        os.environ["webdriver.chrome.driver"]=chromedriver
        self.driver=webdriver.Chrome(chromedriver)
        print("正在打开网页...")
        self.driver.get(url)
```

首先模拟的是进入京东首页，然后在搜索框中输入“手机”进行查询，跳转到相应页面：

```
keyput=self.driver.find_element_by_id("key")
keyput.send_keys(key)
keyput.send_keys(Keys.ENTER)
```

再模拟下拉网页，保证所有手机的数据能够读取到：

```
print("网站正在响应...")
self.driver.implicitly_wait(3)
for i in range(10):
    self.driver.execute_script("var q=document.documentElement.scrollTop={0}".format(i*1000))
    time.sleep(1)
```

网页响应完，获取源代码后，即可定位数据（如何定位的就不多说了，唯一注意的是**放置图片链接的位置有 2 个**，其中一个找不到的话，肯定会在另外一个位置，所以定位了 2 次，再判断哪个不为空）：

```

try:
    phone=self.driver.find_elements_by_xpath("//div[@id='J_goodsList']/li[@class='gl-item']")
    for p in phone:
        try:
            price=p.find_element_by_xpath("//div[@class='p-price']/i").text
            introduce=p.find_element_by_xpath("//div[@class='p-name p-name-type-2']/em").text
            brand=introduce.split(" ")[0]
            brand=brand.replace(",","")#品牌
            quato=introduce.replace(",","")#简介
            evaluate_num=p.find_element_by_xpath("//div[@class='p-commit']/strong/a").text#评价条数
            try:#图片链接
                img1=p.find_element_by_xpath("//div[@class='p-img']/a/img").get_attribute("src")
            except:
                img1=""
            try:
                img2=p.find_element_by_xpath("//div[@class='p-img']/a/img").get_attribute("data-lazy-img")
            except:
                img2=""
            if img1:
                img=img1
            else:
                img=img2
            tlist.append([img,brand,price,quato,evaluate_num])

            img=img2
            tlist.append([img,brand,price,quato,evaluate_num])
        except:
            print("error")
    return tlist
except:
    print("出错")

```

此次爬取，选择爬取**两页**数据，所以定义了**翻页**的方法：

```

def nextPage(self):#翻页
    try:
        previous=self.driver.find_element_by_xpath("//span[@class='p-num']/a[@class='pn-prev disabled']")
    except:
        previous=""
    #只有第一页的上一页按钮的class为“pn-prev disabled”，所以第一页后，previous均为空
    #第一页时，previous不为空，执行点击事件
    if previous:
        next=self.driver.find_element_by_xpath("//span[@class='p-num']/a[@class='pn-next']")
        next.click()

```

如图，只有在第一页能找到 class='pn-prev disabled'的“上一页”属性，找到后，即可进行“下一页”的点击事件。

同样将爬取的数据存入数据库相应的表中：

```

def inputMysql(self,tlist): #数据库先手动建好
    print("正在传入数据库...")
    conn=pymysql.connect(host='localhost',user='root',passwd='admin',db='douban',charset="utf8",cursorclass=pymysql.cursors.DictCursor)
    cur=conn.cursor()
    try:
        for i in tlist:
            cur.execute("insert into jdphone (图片,品牌,价格,简介,评价条数) values (%s,%s,%s,%s,%s)",(i[0],i[1],i[2],i[3],i[4]))
        cur.close()
        conn.commit()
        conn.close()
        print("数据已传入数据库！")
    except:
        print("数据传输失败！")

```

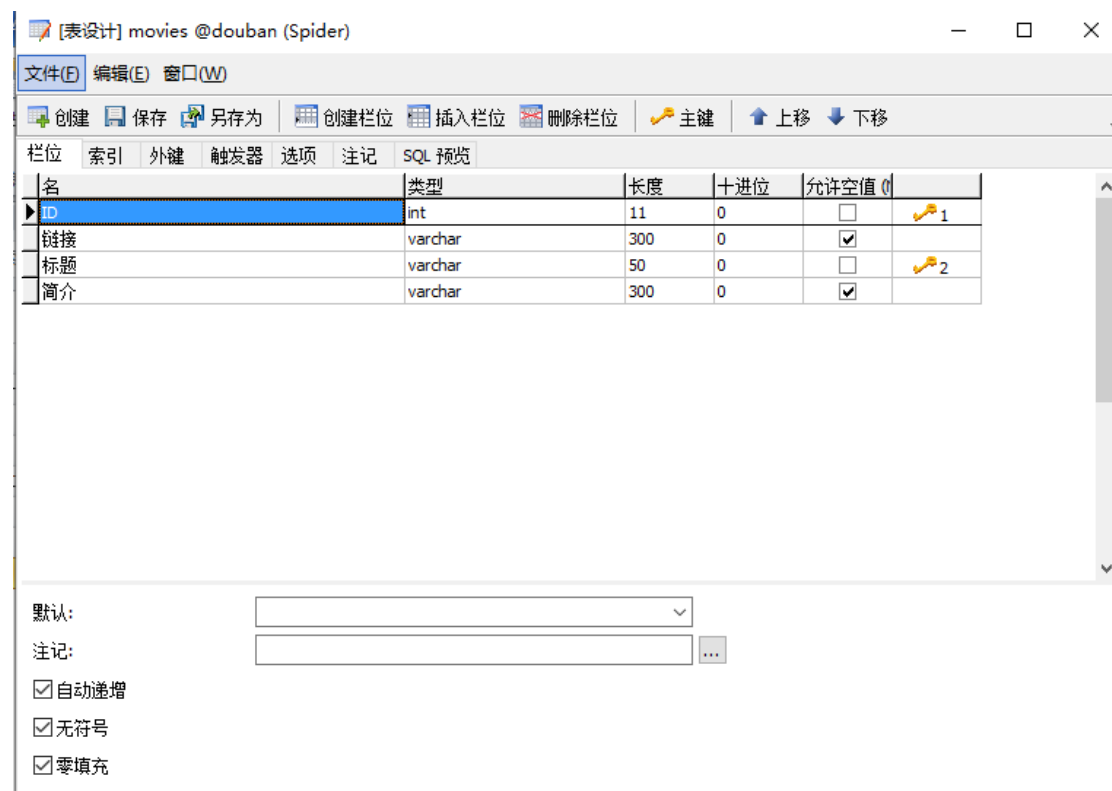
### 3、数据库：（数据库中的两表都是事先创建好的）

```
conn=pymysql.connect(host='localhost',user='root',passwd='admin',db='douban',charset="utf8",cursorclass=pymysql.cursors.DictCursor)
```

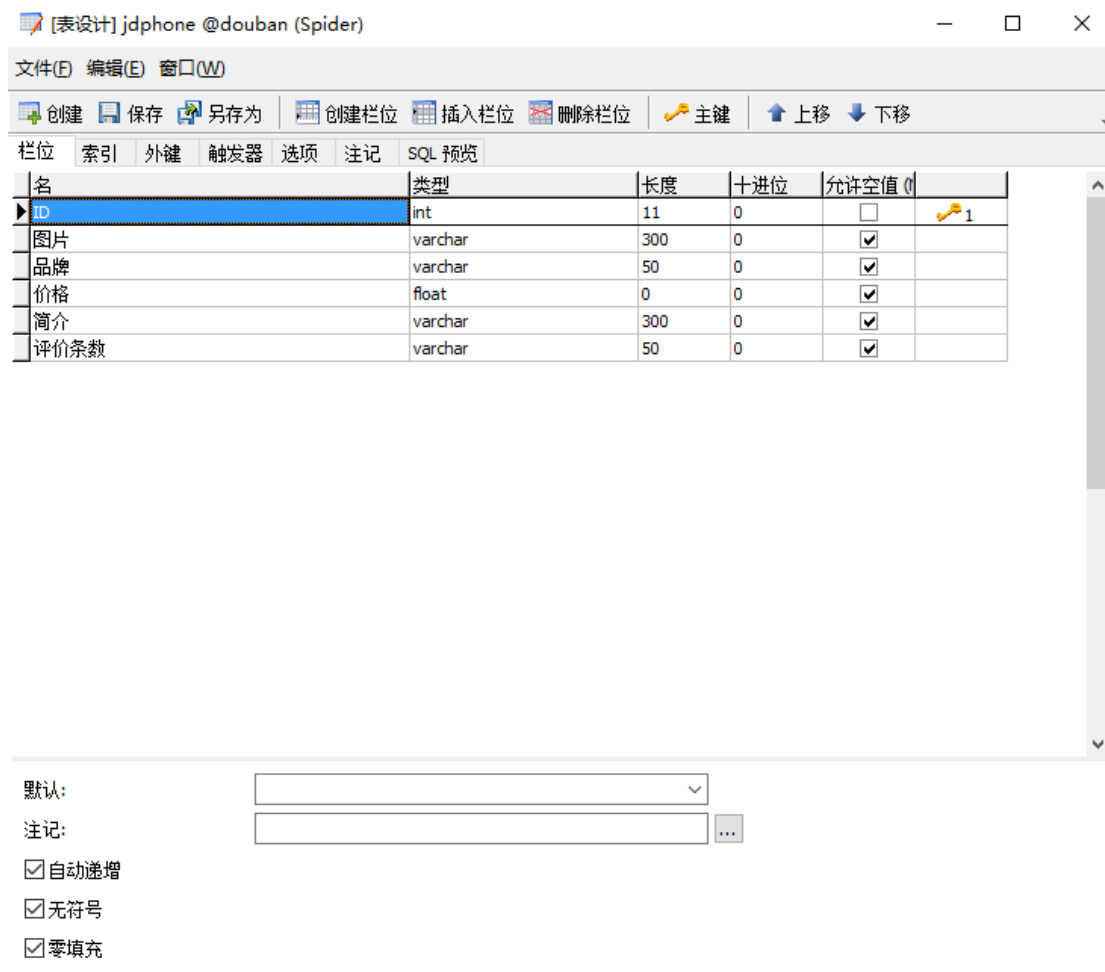
静态爬取数据在 movies，动态爬取数据在 jdphone：



Movies:



Jdphone:

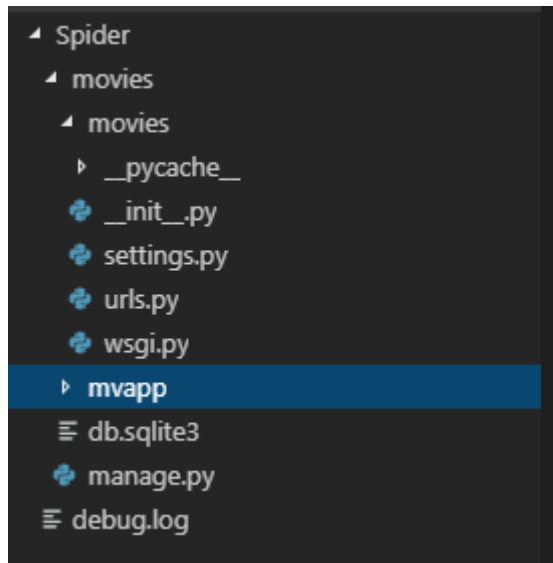


#### 4、Django: (我将静态和动态的代码均放在同一个框架内，所以文件命名可以无视)

使用 `django-admin.py` 来创建 `movies` 项目:

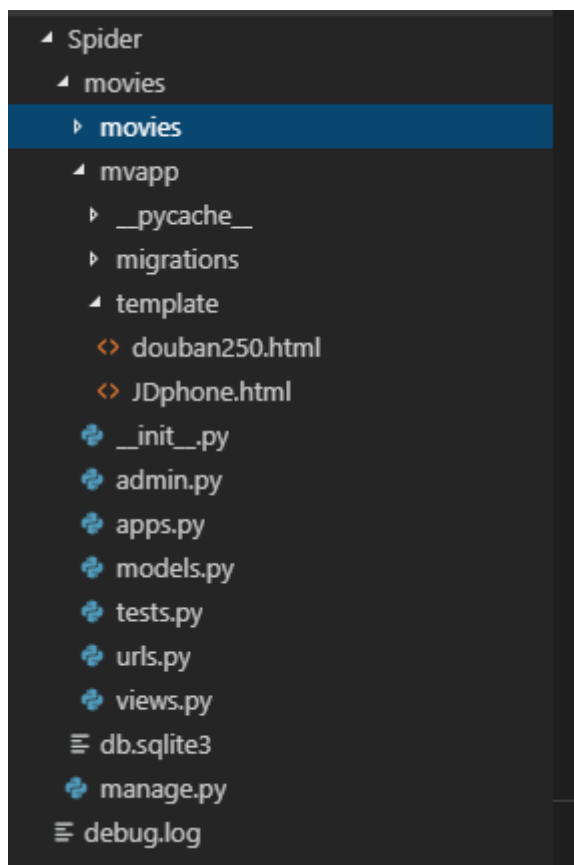
**`django-admin.py startproject movies`**





创建一个 app，来使用模型：

**python manage.py startapp mvapp**



在 mvapp 中新建 template 文件夹，在其中新建 2 个 html 文件分别展示静态和动态爬取的数据：（部分代码）

```

template
douban250.html
JDphone.html

```

```

<body>
<h1 align="center">爬取数据展示区（静态）</h1>
<br>
<table class="douban">
<tr>
<th>ID</th>
<th>链接</th>
<th>标题</th>
<th>简介</th>
</tr>
{% for line in Show%}
<tr>
<td>{{forloop.counter}}</td>
<td>{{line.链接}}</td>
<td>{{line.标题}}</td>
<td>{{line.简介}}</td>
</tr>
{% endfor %}
</table>
</body>

```

```

27 <body>
28 <h1 align="center">爬取数据展示区（动态）</h1>
29 <br>
30 <table class="JD">
31 <tr>
32 <th>ID</th>
33 <th>图片</th>
34 <th>品牌</th>
35 <th>价格</th>
36 <th>简介</th>
37 <th>评价条数</th>
38 </tr>
39 {% for line in JD%}
40 <tr>
41 <td>{{forloop.counter}}</td>
42 <td></td>
43 <td>{{line.品牌}}</td>
44 <td>¥{{line.价格}}</td>
45 <td>{{line.简介}}</td>
46 <td>{{line.评价条数}}</td>
47 </tr>
48 {% endfor %}
49 </table>
50 </body>

```

开始修改相应文件：

Settings.py:

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'mvapp' #添加相应应用
]

```

```

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['*']

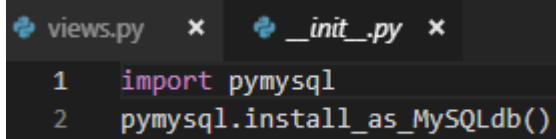
```

Movies 中的 urls.py:

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('Show', views.Show),
    path('JD', views.JD),
]
```

Mvapp 中的\_\_init\_\_.py:



```
1 import pymysql
2 pymysql.install_as_MySQLdb()
```

**Views.py** 中放置了爬取静态、爬取动态、存入数据库、从数据库展示到页面上的所有代码，因代码过多，就不过多展示了：

```

views.py x
1  from django.shortcuts import render
2  import pymysql
3  import requests
4  from bs4 import BeautifulSoup
5
6  from selenium import webdriver
7  from selenium.webdriver.chrome.options import Options
8  from selenium.webdriver.common.keys import Keys
9  import time
10 import os
11
12 # Create your views here.
13
14 #静态爬取豆瓣top250
15 #爬取数据，并存入数据库
16 def getHTMLText(url):
17     try:
18         res=requests.get(url,timeout=60)
19         res.raise_for_status()
20         res.encoding=res.apparent_encoding
21         return res.text
22     except:
23         return "产生异常! "
24
25 def fillList(ulist,html):
26     soup=BeautifulSoup(html,"html.parser")
27     movies=soup.find('ol',class_="grid_view")
28     if movies:
29         for mv in movies.find_all('li'):
30             href=mv.find('div',class_="hd").a.attrs['href']
31             title=mv.find('div',class_="hd").a.span.get_text()
32             quote=mv.find('p',class_="quote").span.get_text()
33             ulist.append([href,title,quote])
34     return ulist

```

运行的总函数如下：

```

def _main():
    mv=[]
    movies=[]
    start_num=0
    for i in range(2):
        url='https://movie.douban.com/top250?start=%s&filter=%s'%(start_num+25*i)
        html=getHTMLText(url)
        movies=fillList(mv,html)
        inputMysql(movies)

#读取数据库，展示在页面上
def get_data(sql):
    conn=pymysql.connect('localhost','root','admin','douban',port=3306,charset="utf8",cursorclass=pymysql.curs
    cur=conn.cursor()
    cur.execute(sql)
    results=cur.fetchall()
    cur.close()
    conn.close()
    return results

def Show(request):
    _main()
    sql="select * from movies"
    movie_list=get_data(sql)
    return render(request,'douban250.html',{'Show':movie_list})
#静态爬取结束

```

```

def Spider(self,url,key,tlist):#运行总过程
    print("开始爬取")
    self.Open_web(url,key)
    phone1=self.fillList(tlist)
    tlist=[]
    self.nextPage()
    print("正在爬取第二页...")
    phone2=self.fillList(tlist)
    self.driver.close()
    self.inputMysql(phone1+phone2)

def JD(request):
    tlist=[]
    url="https://www.jd.com"
    phone=JDphone()
    phone.Spider(url,"手机",tlist)
    sql="select * from jdphone"
    phone_list=get_data(sql)
    return render(request,'JDphone.html',{'JD':phone_list})
#动态爬取结束

```

### 三、运行截图

#### 1、运行前的注意事项：

(1)因代码中并未设计创建数据库的代码，所以数据库必须事先自己建好，上传文件

中应该有相应的 sql 运行文件，**请先导入**；

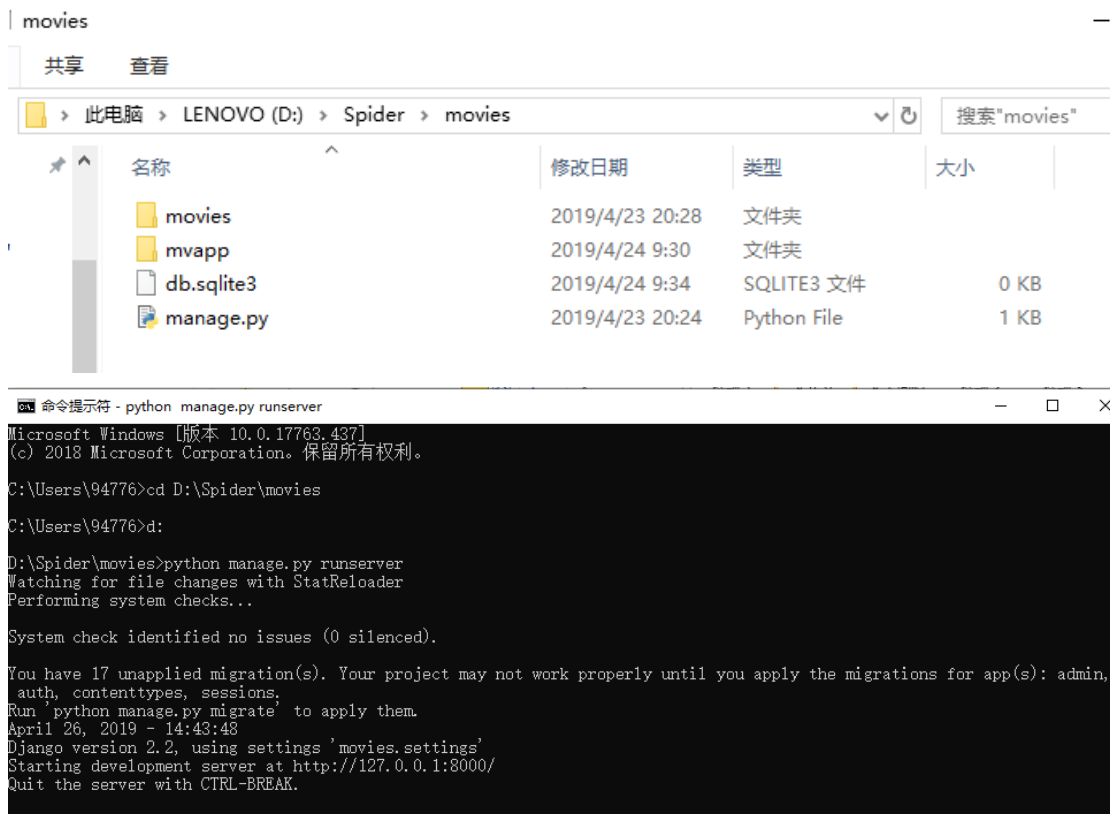
(2) 2 个插入数据库的代码中，会有

“conn=pymysql.connect(host='localhost',user='root',passwd='admin',db='douban',charset="utf8",cursorclass=pymysql.cursors.DictCursor)”，请根据自己创建的数据库，**修改相应的数据**；

(3) 因为 chrome 的模拟浏览器插件在代码中是自定义的路径，所以请先**修改代码中的“chromedriver”**，代码位置如图：

```
class JDphone():
    def Open_web(self,url,key):      #找到查询板块，模拟浏览器输入“手机”搜索手机页面
        chromedriver="D:/平时文件/爬虫/京东手机/chromedriver.exe"
        os.environ["webdriver.chrome.driver"]=chromedriver
        self.driver=webdriver.Chrome(chromedriver)
        print("正在打开网页...")
        self.driver.get(url)
        keyput=self.driver.find_element_by_id("key")
        keyput.send_keys(key)
        keyput.send_keys(Keys.ENTER)
```

## 2、截图：



输入网址: <http://127.0.0.1:8000/Show>

豆瓣电影TOP250 (静态爬取)

127.0.0.1:8000/Show

应用 哔哩哔哩 (゜-゜)つ... AcFun弹幕视频网... 哔哩哔哩, 这里是... 虎牙直播-技术驱动... 大字 游戏

### 爬取数据展示区 (静态)

ID	链接	标题	简介
1	<a href="https://movie.douban.com/subject/1292052/">https://movie.douban.com/subject/1292052/</a>	肖申克的救赎	希望让人自由。
2	<a href="https://movie.douban.com/subject/1291546/">https://movie.douban.com/subject/1291546/</a>	霸王别姬	风华绝代。
3	<a href="https://movie.douban.com/subject/1295644/">https://movie.douban.com/subject/1295644/</a>	这个杀手不太冷	怪蜀黍和小萝莉不得不说的故事。
4	<a href="https://movie.douban.com/subject/1292720/">https://movie.douban.com/subject/1292720/</a>	阿甘正传	一部美国近现代史。
5	<a href="https://movie.douban.com/subject/1292063/">https://movie.douban.com/subject/1292063/</a>	美丽人生	最美的谎言。
6	<a href="https://movie.douban.com/subject/1292722/">https://movie.douban.com/subject/1292722/</a>	泰坦尼克号	失去的才是永恒的。
7	<a href="https://movie.douban.com/subject/1291561/">https://movie.douban.com/subject/1291561/</a>	千与千寻	最好的宫崎骏, 最好的久石让。
8	<a href="https://movie.douban.com/subject/1295124/">https://movie.douban.com/subject/1295124/</a>	辛德勒的名单	拯救一个人, 就是拯救整个世界。
9	<a href="https://movie.douban.com/subject/3541415/">https://movie.douban.com/subject/3541415/</a>	盗梦空间	诺兰给了我们一场无法盗取的梦。
10	<a href="https://movie.douban.com/subject/3011091/">https://movie.douban.com/subject/3011091/</a>	忠犬八公的故事	永远都不能忘记你所爱的人。
11	<a href="https://movie.douban.com/subject/2131459/">https://movie.douban.com/subject/2131459/</a>	机器人总动员	小瓦力, 大人生。
12	<a href="https://movie.douban.com/subject/3793023/">https://movie.douban.com/subject/3793023/</a>	三傻大闹宝莱坞	英俊版憨豆, 高情商版谢耳朵。

[表] movies @douban (Spider)

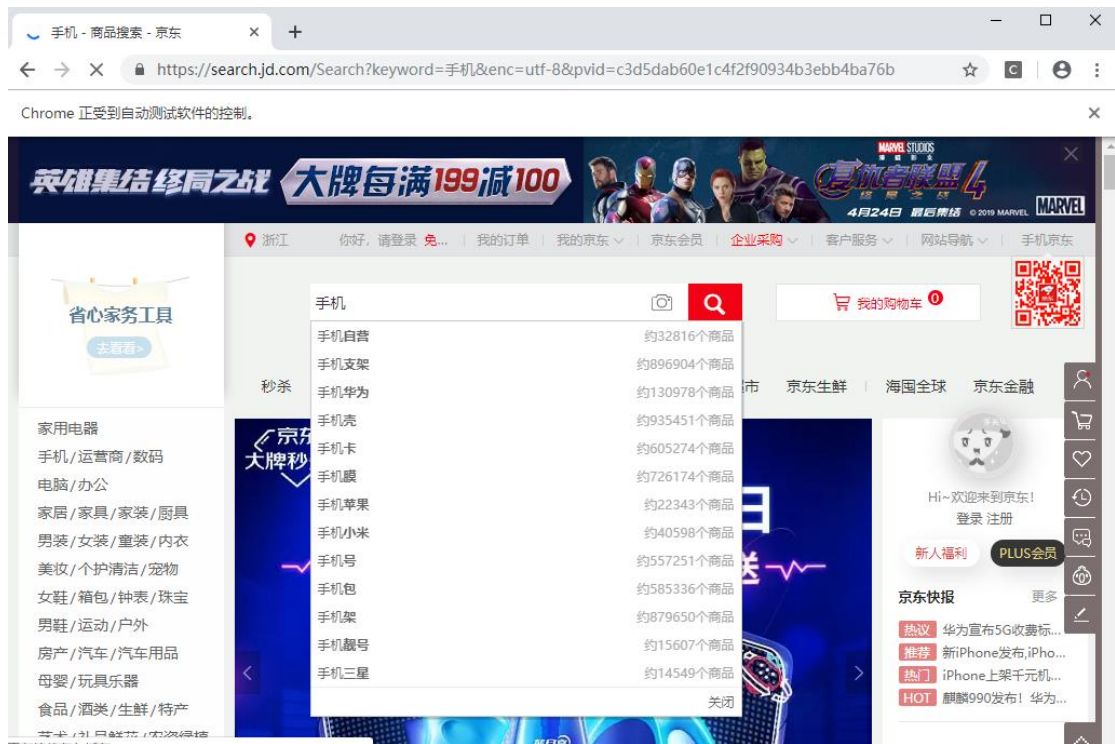
文件(F) 编辑(E) 查看(V) 窗口(W)

导入向导(I) 导出向导(O) 筛选向导 网络视图 表单视图 备注 十六进制 图像 升幂排序

ID	链接	标题	简介
1	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	肖申克的救赎	希望让人自由。
2	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	霸王别姬	风华绝代。
3	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	这个杀手不太冷	怪蜀黍和小萝莉不得不说的故事。
4	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	阿甘正传	一部美国近现代史。
5	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	美丽人生	最美的谎言。
6	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	泰坦尼克号	失去的才是永恒的。
7	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	千与千寻	最好的宫崎骏, 最好的久石让。
8	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	辛德勒的名单	拯救一个人, 就是拯救整个世界。
9	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	盗梦空间	诺兰给了我们一场无法盗取的梦。
10	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	忠犬八公的故事	永远都不能忘记你所爱的人。
11	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	机器人总动员	小瓦力, 大人生。
12	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	三傻大闹宝莱坞	英俊版憨豆, 高情商版谢耳朵。
13	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	海上钢琴师	每个人都要走一条自己坚定了的路, 就算是
14	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	放牛班的春天	天籁一般的童声, 是最接近上帝的存在。
15	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	楚门的世界	如果再也不能见到你, 祝你早安, 午安, 晚
16	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	大话西游之大圣娶亲	一生所爱。
17	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	星际穿越	爱是一种力量, 让我们超越时空感知它的存
18	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	龙猫	人人心中都有个龙猫, 童年就永远不会消失
19	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	教父	千万不要记恨你的对手, 这样会让你失去理
20	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	熔炉	我们一路奋战不是为了改变世界, 而是为了
21	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	无间道	香港电影史上永不过时的杰作。
22	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	疯狂动物城	迪士尼给我们营造的乌托邦就是这样, 永远
23	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	当幸福来敲门	平民励志片。
24	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	怦然心动	真正的幸福是来自内心深处。
25	<a href="https://movie.douban.com/sut">https://movie.douban.com/sut</a>	触不可及	满满温情的高雅喜剧。

输入网址: <http://127.0.0.1:8000/JD>

会开始模拟浏览器操作, 此次操作可能需要点时间



ID	图片	品牌	价格	简介	评价条数
1		【预售】魅族	¥ 3198.0	【预售】魅族 16s 骁龙855全面屏拍照游戏手机 6GB+128GB 碳纤维 全网通移动联通电信4G双卡双待	0
2		荣耀8X	¥ 1298.0	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	145万+



[表] jdphone @douban (Spider)

文件(F) 编辑(E) 查看(V) 窗口(W)

导入向导(I) 导出向导(O) 筛选向导 网格视图 表单视图 备注 十六进位 图像 升幂排序

ID	图片	品牌	价格	简介	评价条数
1	https://img11.360buyimg.com	【预售】魅族	3198	【预售】魅族 16s 骁龙855全面	0
2	https://img14.360buyimg.com	荣耀8X	1298	荣耀8X 千元屏霸 91%屏占比 2K	145万+
3	https://img10.360buyimg.com	Apple	5699	Apple iPhone XR (A2108) 128GB	95万+
4	https://img12.360buyimg.com	【KPL官方比赛用机】vivo	3298	【KPL官方比赛用机】vivo iQOO	9.6万+
5	https://img14.360buyimg.com	荣耀10青春版	1298	荣耀10青春版 幻彩渐变 2400万	49万+
6	https://img12.360buyimg.com	荣耀畅玩8C两天一充	898	荣耀畅玩8C两天一充 莱茵护眼	42万+
7	https://img13.360buyimg.com	小米	1199	小米 红米Redmi Note7 幻彩渐变	60万+
8	https://img10.360buyimg.com	vivo	799	vivo U1 水滴全面屏 AI智慧拍照	9.8万+
9	https://img10.360buyimg.com	OPPO	3599	OPPO Reno 全面屏拍照手机 8G	9200+
10	https://img13.360buyimg.com	荣耀V20	2798	荣耀V20 胡歌同款 麒麟980芯片	24万+
11	https://img10.360buyimg.com	小米9	3299	小米9 4800万超广角三摄 8GB+	14万+
12	https://img11.360buyimg.com	小米8SE	1399	小米8SE 全面屏智能游戏拍照手	68万+
13	https://img10.360buyimg.com	小米8青春版	1499	小米8青春版 镜面渐变AI双摄 6	29万+
14	https://img12.360buyimg.com	小米	799	小米 红米Redmi 7 AI双摄 3GB+	7.6万+
15	https://img14.360buyimg.com	vivo	1598	vivo Z3 6GB+64GB 极光蓝 性能3	35万+
16	https://img13.360buyimg.com	黑鲨游戏手机2	3499	黑鲨游戏手机2 8GB+128GB 暗影	4.5万+
17	https://img13.360buyimg.com	联想Z6	2999	联想Z6 Pro 8GB+128GB 黑色 骁龙	0
18	https://img13.360buyimg.com	Apple	6099	Apple iPhone X (A1865) 64GB 深空	131万+
19	https://img11.360buyimg.com	vivo	2298	vivo S1 6GB+128GB 冰湖蓝 2480	8200+
20	https://img12.360buyimg.com	荣耀10	2198	荣耀10 GT游戏加速 AIS手持夜视	117万+
21	https://img11.360buyimg.com	vivo	3598	vivo X27 8GB+256GB大内存 雀羽	2.7万+
22	https://img10.360buyimg.com	小米	1599	小米 红米Redmi Note7Pro AI双摄	7.7万+
23	https://img13.360buyimg.com	小米	649	小米 红米6A AI美颜 3GB+32GB	82万+
24	https://img12.360buyimg.com	Apple	3799	Apple iPhone 8 (A1863) 64GB 深空	131万+
25	https://img12.360buyimg.com	荣耀20i	1599	荣耀20i 3200万AI自拍 超广角三	9100+
26	https://img10.360buyimg.com	Apple	4699	Apple iPhone 8 Plus (A1864) 64GB	178万+

## 四、未解决的问题

设计中，出现的一些至今还未解决的明显问题：

- 1、完整的执行一次代码后，再执行一次代码，同样的数据会再次存入数据库，数据会不断重复、叠加。比如第一次爬取了第一页数据，第二次爬取了第一、二页的数据，都存入数据库后，会导致第一页的数据出现了2次。（目前解决办法：每次执行代码前，将数据库的数据先全部清空）
- 2、数据展示页面的展示记录过多，未实现分页功能。
- 3、设计框架和美化能力有限。