

温州大學瓯江學院

WENZHOU UNIVERSITY OUJIANG COLLEGE

《爬虫与数据分析》期末作业

题 目: 爬虫期末大作业

二级学院: 数信学院

班 级: 16 计算机科学与技术三班

姓 名: 李雪萌

学 号: 16219111337

完成日期: 2019. 06. 17

温州大学瓯江学院教务部

二〇一二年十一月制

项目内容:

Django 项目+爬虫代码+数据库+实验报告+12306 登陆验证

涉及知识:

Python 语言的基本语法以及第三方库的使用;

静态网页的爬取;

网页的解析;

动态网页的爬取;

模拟无头浏览器爬取动态网页;

爬虫与反爬虫

登陆与自动验证

涉及其他:

正则表达式

Django 的 mvc 模型

Selenium 的安装调用

豆瓣爬取:

```
1 import requests
2 from lxml import etree
3 import MySQLdb
4
5
6 con= MySQLdb.connect(host='localhost',user='root',password='root',db='test',charset="utf8")
7 cursor=con.cursor()
8 cursor.execute("create table test(No varchar(32) ,content char(255))")
9
10 #静态网页抓取: request爬虫实践: top250电影数据
11 def get_page(start_num):
12     url='https://movie.douban.com/top250?start=%s&filter=' %start_num
13     print(url)
14
15     response=requests.get(url)
16     tree=etree.HTML(response.text)
17     title=tree.xpath('//span[@class="title"]')[1]/text()
18     return title
19
20 def get_all_page(start,end):
21     result=[]
22     for i in range(start,end-start):
23         title_list=get_page(i*25)
24         result+=title_list
25         print(result)
26     return result
27
28 if __name__=="__main__":
29     result=get_all_page(0,10)
30
31     for i in range(len(result)):
32         cursor.execute("INSERT INTO test(No,content) values(%d,'%s')" %(i+1,result[i]))
33     cursor.close()
34     con.commit()
35     con.close()
```

天气爬取:

```

from bs4 import BeautifulSoup
from bs4 import UnicodeDammit
import urllib.request
import MySQLdb

class weatherDB:
    def openDB(self):
        self.con=MySQLdb.connect(host='localhost',user='root',password='root',db='test',charset="utf8")
        self.cursor=self.con.cursor()
        try:
            self.cursor.execute("create table weathers1(wCity varchar(16),wDate varchar(16),wWeather varchar(64),wTemp varchar(32),constraint pk_weather primary key(wCity,wDate))")
        except:
            self.cursor.execute("delete from weathers")

    def closeDB(self):
        self.con.commit()
        self.con.close()

    def insert(self,city,date,weather,temp):
        try:
            self.cursor.execute("insert into weathers1(wCity,wDate,wWeather,wTemp) values(?, ?, ?, ?)",(city,date,weather,temp))
        except Exception as err:
            print(err)

    def show(self):
        self.cursor.execute("select * from weathers1")
        rows=self.cursor.fetchall()
        print("%-16s%-16s%-32s%-16s" % ("city","date","weather","temp"))
        for row in rows:
            print("%-16s%-16s%-32s%-16s" % (row[0],row[1],row[2],row[3]))


class weatherForecast:
    def __init__(self):
        self.headers={"User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36"}
        self.cityCode={"北京": "101010100", "上海": "101020100", "广州": "101280101", "深圳": "101280601"}

    def forecastCity(self,city):
        if city not in self.cityCode.keys():
            print(city+" code cannot be found")
        return

    url="http://www.weather.com.cn/weather/"+self.cityCode[city]+".html"
    try:
        req=urllib.request.Request(url,headers=self.headers)
        data=urllib.request.urlopen(req)
        data=data.read()
        dammit=UnicodeDammit(data,['utf-8','gbk'])
        data=dammit.unicode_markup
        soup=BeautifulSoup(data,"lxml")
        lis=soup.select("ul[class='t clearfix'] li")
        for li in lis:
            try:
                date=li.select('h1')[0].text
                weather=li.select('p[class="wea"]')[0].text
                temp=li.select('p[class="tem"] span')[0].text+"/"+li.select('p[class="tem"] i')[0].text
                print(city,data,weather,temp)
                self.db.insert(city,date,weather,temp)
            except Exception as err:
                print(err)
    except Exception as err:
        print(err)

    def process(self,cities):
        self.db=weatherDB()
        self.db.openDB()
        for city in cities:
            self.forecastCity(city)

        #self.db.show()
        self.db.closeDB()

ws=weatherForecast()
ws.process(["北京", "上海", "深圳", "广州"])
print("completed")

```

京东爬取：

```

1  from selenium import webdriver
2  from selenium.webdriver.chrome.options import Options
3  import urllib.request
4  import threading
5  import MySQLdb
6  import os
7  import datetime
8
9
10
11 class MySpider:
12     header={"User-Agent: Mozilla/5 (Windows NT 10 ; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36"}
13     imagePath="download"
14
15     def startUp(self,url,key):
16         try:
17             self.con=MySQLdb.connect(host='localhost',user='root',password='root',db='test',charset="utf8")
18             self.cursor=self.con.cursor()
19             try:
20                 self.cursor.execute("drop table phones")
21
22             except:
23                 pass
24             try:
25                 sql="create table phones(mNo varcae(32) primary key,mMark varchar(256),mPrice varchar(32),mNote varchar(1024),mFile varchar(256))"
26                 self.cursor.execute(sql)
27             except:
28                 pass
29             except Exception as err:
30                 print(err)
31
32             chrome_option=Options()
33             chrome_option.add_argument('--headless')
34             chrome_option.add_argument('--disable-gpu')
35
36             self.driver=webdriver.Chrome(chrome_options=chrome_option)
37             self.threads=[]
38             self.No=0
39             self.imgNo=0
40
41             try:
42                 if not os.path.exists(MySpider.imagePath):
43                     os.mkdir(MySpider.imagePath)
44                 images=os.listdir(MySpider.imagePath)
45                 for img in images:
46
47                     for img in images:
48                         s=os.path.join(MySpider.imagePath,img)
49                         os.remove(s)
50
51             except Exception as err:
52                 print(err)
53             try:
54                 if not os.path.exists(MySpider.imagePath):
55                     os.mkdir(MySpider.imagePath)
56                 images=os.listdir(MySpider.imagePath)
57                 for img in images:
58                     s=os.path.join(MySpider.imagePath,img)
59                     os.remove(s)
60             except Exception as err:
61                 print(err)
62             self.driver.get(url)
63             keyInput=self.driver.find_element_by_id("key")
64             keyInput.send_keys(key)
65             keyInput.send_keys(Keys.ENTER)
66
67         def closeUp(self):
68             try:
69                 self.con.commit()
70                 self.con.close()
71                 self.driver.close()
72             except Exception as err:
73                 print(err)
74
75         def insertDB(self,mNo,mMark,mPrice,mNote,mFile):
76             try:
77                 sql="insert into phones(mNo,mMark,mPrice,mNote,mFile)values(?,?,?,?,?)"
78                 self.cursor.execute(sql,(mNo,mMark,mPrice,mNote,mFile))
79             except Exception as err:
80                 print(err)
81
82         def showDB(self):
83             try:
84                 con= MySQLdb.connect(host='localhost',user='root',password='root',db='test',charset="utf8")
85                 cursor=con.cursor()
86                 print("%-8s %-16s %-8s %-16s %s%("No","Mark","Price","Image","Note"))
87                 cursor.execute("select mNo,mMark,mPrice,mNote from phones order by mNo")
88                 rows=cursor.fetchall()
89                 for row in rows:
90                     print("%-8s %-16s %-8s %-16s %s%(%(row[0],row[1],row[2],row[3],row[4]))")
91                 con.close()
92

```

```

        close()
    except Exception as err:
        print(err)
def download(self,src1,src2,mFile):
    data=None
    if src1:
        try:
            req=urllib.request.Request(src1,headers=MySpider.headers)
            resp=urllib.request.urlopen(req,timeout=400)
            data=resp.read()
        except:
            pass
    if not data and src2:
        try:
            req=urllib.request.Request(src2,headers=MySpider.headers)
            resp=urllib.request.urlopen(req,timeout=400)
            data=resp.read()
        except:
            pass
    if data:
        fobj=open(MySpider.imagePath+"\\"+mFile,"wb")
        fobj.write(data)
        fobj.close()
        print("download",mFile)

def processSpider(self):
    try:
        time.sleep(10)
        print(self.driver.current_url)
        lis=self.driver.find_element_by_xpath("//div[@id='J_goodsList']//li[@class='gl-item']")
        for li in lis:
            try:
                src1=li.find_element_by_xpath(".\\div[@class='p-img']//a//img").get_attribute("src")
            except:
                src1=""
            try:
                src2=li.find_element_by_xpath(".\\div[@class='p-img']//a//img").get_attribute("data-lazy-img")
            except:
                price="0"
            try:
                note=li.find_element_by_xpath("//div[@class='p-price']//i").text
                mark=note.split("")[0]
                mark=mark.replace("爱心冻东\n","")
                mark=mark.replace(",","");
                note=note.replace("爱心冻东\n","");
                note=note.replace(",","");
                note=note.replace(".", "")
            except:
                note=""
                mark=""
            self.No+=self.No+1
            no=str(self.No)
            while len(no)<6:
                no="0"+no
            print(no,mark,price)
            if src1:
                src1=urllib.request.urljoin(self.driver.current_url,src1)
                p=src1.rfind("/")
                mFile=no+src1[p:]
            elif src2:
                src2=urllib.request.urljoin(self.driver.current_url,src2)
                p=src2.rfind("/")
                mFile=no+src2[p:]
            if src1 or src2:
                T=threading.Thread(target=self.download,args=(src1,src2,mFile))
                T.setDaemon(False)
                T.start()
                self.threads.append(T)
            else:
                mFile=""
                self.insertDB(no,mark,price,no,mFile)
        try:
            self.driver.find_element_by_xpath("//span[@class='p-num']//a[@class='pn-next disabled']")
        except:
            nextPage=self.driver.find_element_by_xpath("//span[@class='p-num']//a[@class='pn-next']")
            nextPage.click()
            self.processSpider()
    except Exception as err:
        print(err)

def executeSpider(self,url,key):
    starttime=datetime.datetime.now()
    print("Spider starting.....")
    self.startUp(url,key)
    self.processSpider()
    self.closeUp()
    for t in self.threads:
        t.join()
    print("Spider completed.....")
    endtime=datetime.datetime.now()
    elapsed=(endtime-starttime).seconds
    print("Total",elapsed,"second elapsed")

```

```
t.join()
print("Spider completed.....")
endtime=datetime.datetime.now()
elapsed=(endtime-starttime).seconds
print("Total",elapsed,"second elapsed")

url='http://www.jd.com'
spider=MySpider()
while True:
    print("1.爬取")
    print("2.显示")
    print("3.退出")
    s=input("请输入选择 (1, 2, 3) :")
    if s=="1":
        spider.executeSpider(url,"手机")
    elif s=="2":
        spider.showDB()
    elif s=="3":
        break
```

淘宝爬取：

```
import requests
import re
import MySQLdb

def getHTMLText(url):
    try:
        r=requests.get(url,timeout=30)
        r.raise_for_status()
        r.encoding=r.apparent_encoding
        return r.text
    except:
        return ""

def parsePage(ilt,html):
    try:
        plt=re.findall(r"\\"view_price\"\:\\"[\d\.]*\"",html)
        tlt=re.findall(r"\\"raw_title\"\:\\".*?\"",html)
        for i in range(len(plt)):
            price=eval(plt[i].split(':')[1])
            title=eval(tlt[i].split(':')[1])
            ilt.append([price,title])
    except:
        print("")

def printGoodList(ilt):
    con=MySQLdb.connect(host='localhost',user='root',password='root',db='test',charset="utf8")
    cursor=con.cursor()
    cursor.execute("select * from shubao")
    tplt="{:.4}\t{:.8}\t{:.16}"
    print(tplt.format("序号","价格","商品名称"))
    count=0
    for g in ilt:
        count+=1
        print(tplt.format(count,g[0],g[1]))
        cursor.execute("insert into shubao(id,Price,Title) values('"+count+"','"+g[0]+"','"+g[1]+"')")
    cursor.close()
    con.commit()
    con.close()

def main():
    goods='背包'
    depth=3

    start_url='https://s.taobao.com/search?q=' + goods
    infolist=[]
    for i in range(depth):
        try:
            url=start_url+'&s='+str(44*i)
            html=getHTMLText(url)
            parsePage(infolist,html)
        except:
            continue
        printGoodList(infolist)
    main()
```

爬取结果展示：

```
e:\爬虫\16219111337_lixuemeng\pythonfile>python tq.py
list index out of range
北京 26日（明天） 晴转多云 21°C~10°C
北京 27日（后天） 多云 18°C~7°C
北京 28日（周日） 多云 21°C~9°C
北京 29日（周一） 多云转小雨 24°C~13°C
北京 30日（周二） 多云 26°C~14°C
北京 1日（周三） 多云 24°C~14°C
list index out of range
上海 26日（明天） 多云转晴 18°C~12°C
上海 27日（后天） 多云 18°C~14°C
上海 28日（周日） 小雨转多云 23°C~17°C
上海 29日（周一） 中雨转阴 24°C~17°C
上海 30日（周二） 阴转小雨 22°C~17°C
上海 1日（周三） 小雨转多云 23°C~17°C
list index out of range
广州 26日（明天） 中雨转中到大雨 28°C~24°C
广州 27日（后天） 中到大雨转雷阵雨 28°C~24°C
广州 28日（周日） 雷阵雨 29°C~25°C
广州 29日（周一） 雷阵雨转中雨 30°C~25°C
广州 30日（周二） 中雨转大到暴雨 30°C~22°C
广州 1日（周三） 大到暴雨转多云 26°C~20°C
list index out of range
深圳 26日（明天） 中雨转大雨 29°C~24°C
深圳 27日（后天） 大雨转雷阵雨 27°C~23°C
深圳 28日（周日） 雷阵雨 28°C~24°C
深圳 29日（周一） 雷阵雨 30°C~25°C
深圳 30日（周二） 雷阵雨转暴雨 30°C~23°C
深圳 1日（周三） 暴雨转阵雨 27°C~23°C
完成!
```

47	289.00	牛津布双肩包女2019新款韩版时尚尼龙书包电脑包双肩旅行防盜背包
48	69.00	双肩背包女背包2018新款韩版潮牛津布帆布时尚百搭书包旅行小包包2019
49	69.00	双肩背包女背包2018新款韩版潮牛津布帆布时尚百搭书包旅行小包包2019
50	59.00	男士背包电脑包休闲韩版时尚潮流高中生书包大容量旅行双肩包
51	29.00	纯棉双肩包男女士双肩包休闲运动背包男女通用书包学生书包
52	239.00	纯棉男双肩包休闲运动背包男女通用书包学生书包
53	149.00	纯棉男双肩包休闲运动背包男女通用书包学生书包
54	169.00	纯棉男双肩包大容量旅行商务休闲背包男时尚潮流牛皮书包
55	699.00	纯棉男双肩包皮质大学生书包休闲青年电脑背包时尚潮流牛皮书包
56	189.00	七匹狼双肩包男 新款商务男士旅行双肩包女休闲大容量电脑书包
57	139.00	赫登尔双肩包男牛津布双肩包男休闲旅行时尚休闲学生电脑包
58	109.00	MOTO米图双肩包男大学生书包休闲旅行包潮流牛皮书包
59	139.00	weak双肩包女休闲学生1-6年级书包男休闲旅行包潮流牛皮书包
60	228.00	anello双肩包女休闲学生书包休闲旅行包潮流牛皮书包aleno定制设计旅行背包休闲双肩包女高中生书包高中生书包牛津布双肩包
61	29.90	定制设计旅行背包休闲双肩包女高中生书包高中生书包牛津布双肩包
62	159.00	双肩背包男士大容量旅行时尚潮流电脑商务办公功能书包男
63	149.00	七匹狼双肩包男双肩包商务电脑包休闲书包时尚潮流牛津布
64	158.00	专柜正品Jansport杰斯双肩背包男女同款学生书包T5058结实耐用
65	89.00	休闲双肩包男女休闲旅行书包高中生书包大容量双肩包男书包
66	116.00	休闲双肩包男女休闲旅行书包高中生书包大容量双肩包男书包
67	149.00	双肩背包男潮流休闲包男学生15.6寸电脑包旅行包男时尚潮流
68	298.00	OK双肩男时尚潮流背包大学生书包休闲潮流牛津布旅行包新品
69	79.00	日本乐天正品earthy背包2019新款书包妈妈包出行高能出行
70	49.00	双肩包男休闲学生书包男双肩包女高中生书包高中生书包初中生书包
71	139.00	途德双肩包男士大容量旅行包休闲旅行时尚潮流牛津布
72	319.00	香港潮牌2019新款双肩包男高中生书包旅行书包时尚潮流牛津布
73	142.00	七匹狼双肩包男高中生书包休闲旅行书包高中生书包
74	158.00	休闲双肩包男潮流休闲包高中生书包大学生书包潮流牛津布
75	119.00	双肩背包男休闲功能商务男包15.6寸电脑包
76	189.00	七匹狼双肩包男潮流休闲包高中生书包高中生书包潮流牛津布
77	249.00	莱夫2019新款双肩背包女帆布大容量商务电脑包牛津布旅行包女包
78	124.00	男士双肩包韩版学生书包男大学生书包潮流牛津布旅行包
79	69.90	途卡休闲背包男双肩包旅行包高中生书包高中生书包
80	29.90	吉普森双肩包男高中生书包高中生书包高中生书包
81	169.00	瑞士军刀双肩包高中生书包初中生书包高中生书包潮流旅行背包女
82	29.00	小米双肩包女大容量书包通用运动包日常休闲背包书包
83	168.00	F4儿童书包小学生书包6-12岁女童书包1-3-6岁防水拉链拖拉
84	129.00	米熙背包男双肩包男大容量学生书包休闲商务电脑包女款潮流旅行包
85	138.00	尼龙背包男时尚休闲包男旅行包电脑包潮流书包高中生书包
86	228.00	anello双肩包女高中生15.6寸电脑包休闲旅行包潮流牛津布
87	59.00	东口袋潮流休闲包男高中生书包高中生书包高中生书包
88	388.00	Snowmax斯诺麦双肩包女高中生书包休闲男双肩包潮流旅行包BP2
89	68.00	迪士尼书包小学书包1-3-4-5年级男童女童书包会发光长便儿童双肩包
90	399.00	F1ION非牛仔双肩包女休闲旅行背包女士拉链印花背包青年时尚书包小包
91	274.62	上海迪士尼包包之菲尔双肩背包帆布潮流休闲旅行大容量登山书包
92	59.00	背包男双肩背包旅行包男轻便旅游行李包休闲时尚大容量登山书包
93	98.00	休闲背包男双肩包时尚潮流青年男女士旅行高中生书包大容量电脑包
94	109.00	书包男双肩包高中生书包高中生书包高中生书包高中生书包
95	169.00	千层乐乐正品休闲时尚双肩包男背包女包潮流旅行背包高中生书包
96	89.00	超火双肩包女2019新款潮流旅行背包休闲包高中生书包高中生书包高中ins风
97	19.90	迪士尼旗舰店双肩包新款男书包旅行运动背包轻便女包QUBI

98	69.00	
99	79.00	从2019年1月1日起白金会员向所有客户(大陆)开具增值税专用发票
100	269.00	高中生书包女双肩包2019新款潮流时尚中背包 高中生书包女双肩包2019新款潮流时尚中背包 高中生书包女双肩包2019新款潮流时尚中背包
101	49.00	高中生书包女2019新款潮流时尚中背包
102	79.90	高中生书包女2019新款潮流时尚中背包
103	159.00	高中生书包女2019新款潮流时尚中背包
104	139.00	高中生书包女2019新款潮流时尚中背包
105	169.00	高中生书包女2019新款潮流时尚中背包
106	69.00	瑞士军刀双肩包女中学生书包休闲商务男士旅行大容量电脑包 瑞士王子王室拉杆箱3-6岁儿童女童拉杆箱3-6岁儿童女童
107	59.00	书包男高中生初中生大学生男背包书包男休闲双肩包时尚潮流男包
108	59.00	书包男高中生初中生大学生男背包书包男休闲双肩包时尚潮流男包
109	129.00	书包男高中生初中生大学生男背包书包男休闲双肩包时尚潮流男包
110	149.00	书包男高中生初中生大学生男背包书包男休闲双肩包时尚潮流男包
111	258.00	Jansport旗舰店官网正品双肩包学生书包女正品女士女背包T50
112	698.00	Herschel Supply Retreat经典时尚潮流男双肩背包书包背包10065
113	38.00	小学生书包男3-4-5岁双肩包减负儿童书包便携男背包
114	1.50	运动防雨抽绳双肩包拉链书包学生书包便携男背包
115	169.90	运动风运动双肩包背包男书包男健身包旅行包休闲商务KIPSTA
116	149.90	运动风运动双肩包背包男书包男健身包旅行包休闲商务
117	118.00	日本TCL LOCAL A 高品质帆布包旅行包单肩斜挎包邮费信包
118	15.90	高中生书包女2019新款潮流女中学生书包单肩包斜挎包
119	298.00	OKE品双肩包男时尚潮流书包大学生书包休闲男包韩版旅行包男包
120	148.00	香港正品doughnut甜甜圈双肩包韩版潮流时尚男女学生书包
121	158.00	韩国正品doughnut甜甜圈双肩包韩版潮流时尚男女学生书包
122	59.00	韩国正品doughnut甜甜圈双肩包韩版潮流时尚男女学生书包
123	35.00	幼儿园书包女童书包男童书包休闲书包
124	49.00	幼儿园书包女童书包男童书包休闲书包
125	99.00	小学生书包4-12周岁 美泰公司 双肩包3-5岁组女童书包 1-3岁组女孩三环包书包男童书包男童书包地主不同款双肩包时尚潮流休闲书包中小学生书包
126	318.00	香港IT双肩包女学生2019新款潮流时尚背包休闲旅行书包容量书包
127	52.00	LEQUEEN防水布书包女双肩包女背包大容量男大容量女高中生书包
128	118.00	随身便携地主学生书包男休闲书包潮流书包男高中生书包
129	169.00	高中生书包女2019新款潮流时尚背包男高中生书包
130	188.00	高中生书包女2019新款潮流时尚背包男高中生书包
131	69.00	高中生书包女3-4-5-6岁书包男3-5-6岁书包男高中生书包
132	179.00	德威双肩包时尚潮流背包男士旅行包韩版休闲青春学生书包电脑包
133	936.00	美国正品J'story burchy'包TB双肩包2017新款书包旅行包背包
134	39.00	ins超火的双肩包韩版背包男高中生书包男电脑包旅行包书包
135	29.90	ins超火的双肩包韩版背包男高中生书包男电脑包旅行包书包
136	888.00	新郎正品 coach蔻驰包包真皮男士商务背包休闲旅行书包电脑双肩包

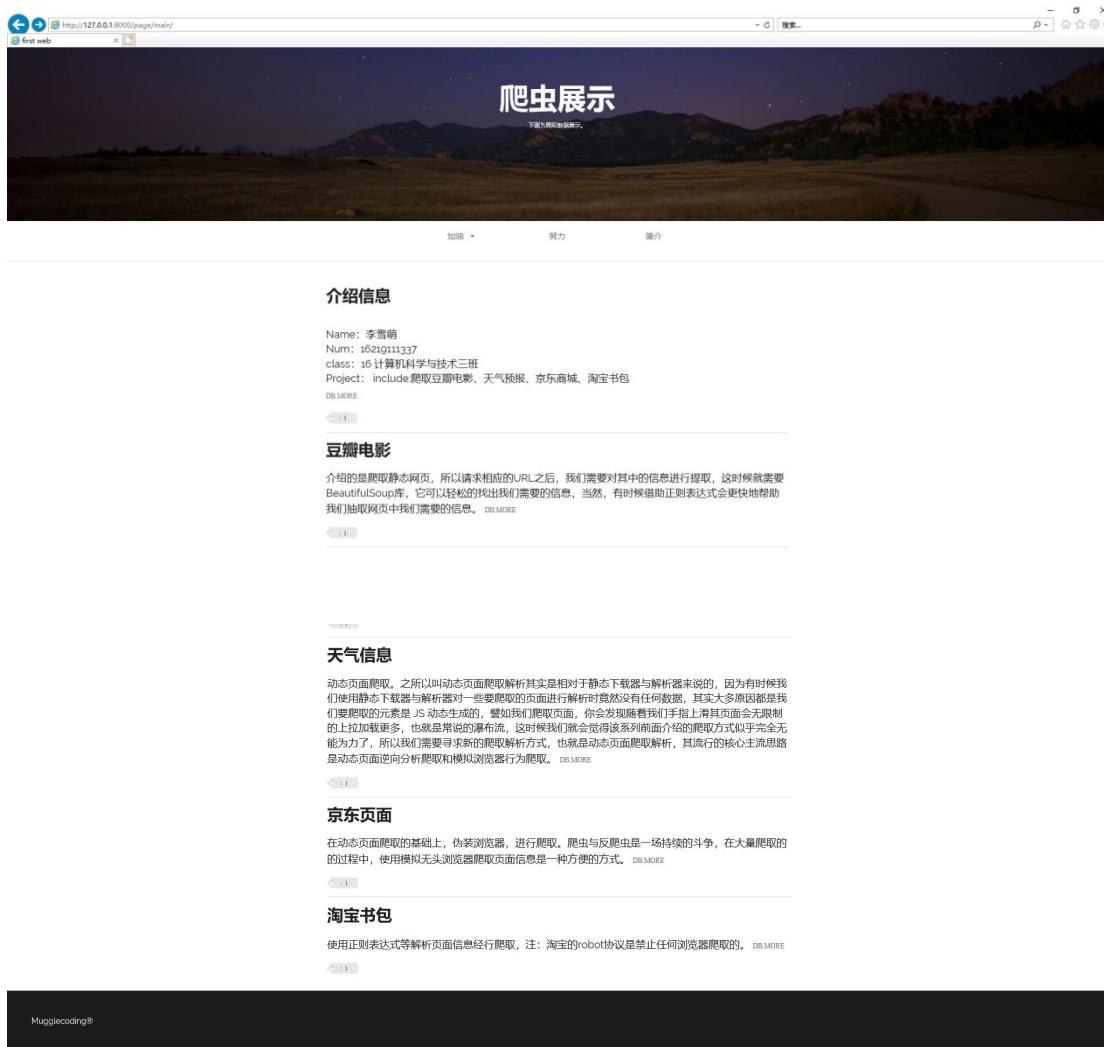
网站思想：

1. 运用 mvc 框架（简单用 mvt）

2. Template 模板页用于显示 view 层的信息

3. Model 层是 django 特有的数据库形式，以类的方式引用

4. 页面使用 html+css (semantic ui) + 简单 js+django 特有编辑语言



代码：

web:

```

1  <!DOCTYPE html>
2  {% load staticfiles %}
3  <html>
4    <head>
5      <meta charset="utf-8">
6      <title>first web</title>
7      <link rel="stylesheet" href="{% static 'css/semantic.css' %}" media="screen" title="no title" charset="utf-8">
8      <link href="https://fonts.googleapis.com/css?family=Oswald|Raleway" rel="stylesheet">
9
10
11      <style type="text/css">
12          h1 {
13              font-family:'Oswald', sans-serif!important;
14              font-size:40px;
15          }
16
17          body {
18              font-family: 'Raleway', sans-serif;
19          }
20          p {
21              font-family: 'Raleway', sans-serif;
22              font-size:18px;
23          }
24
25          .ui.vertical.segment.masthead {
26              height: 300px;
27              background-image: url("{% static 'images/star_banner.jpg' %}");
28              background-size: cover;
29              background-position: 100% 80%;
30          }
31
32          .ui.container.segment {
33              width: 800px;
34          }
35
36          .ui.center.aligned.header.blogslogon {
37              margin-top: 40px;
38          }
39
40          .ui.center.aligned.header.blogslogon p {
41              margin-top: 10px;
42              color: white;
43              font-size: 10px;
44          }
45          .ui.container.nav {
46              width: 500px;
47          }
48
49      </style>
50      {% block css %}{% endblock%}
51
52  </head>
53  <body>
54      <div class="ui inverted vertical segment masthead">
55
56          <h1 class="ui center aligned header blogslogon" style="font-size:50px;font-family: 'Raleway', sans-serif!important;">
57              跳虫展示
58              <p class="ui sub header">
59                  | 下面为爬虫数据展示。
60              </p>
61
62          </h1>
63      </div>
64      <div class="ui container nav">
65          <div class="ui borderless text three item menu">
66              <div class="ui simple dropdown item">
67                  加油
68                  <i class="dropdown icon"></i>
69                  <div class="menu">
70                      <a class="item" href="">简介</a>
71                      <a class="item" href="">简介</a>
72                  </div>
73              </div>
74              <a class="item">
75                  | 努力
76              </a>
77              <a class="item">
78                  | 简介
79              </a>
80
81          </div>
82      </div>
83
84      <div class="ui divider"></div>
85
86      {% block header %}{% endblock%}
87
88      <div class="ui vertical segment">
89          <div class="ui container vertical segment">
90              <a href="#">
91                  <h1 class="ui header">
92                      | 介绍信息
93                  </h1>
94              </a>
95
96          </div>
97      </div>

```

```
</a>
<i class="icon grey small unhide"></i>
<p>
    <form>
        <div class="six wide column">
            <p><label>Name: 李雪萌</label></p>
        </div>
        <div>
            <p><label>Num: 16219111337</label></p>
        </div>
        <div>
            <p><label>class: 16 计算机科学与技术三班</label></p>
        </div>
        <div>
            <p>
                <label>
                    Project:
                    include:爬取豆瓣电影、天气预报、京东商城、淘宝书包
                </label>
            </p>
        </div>
    </form>
    <a href="#">
        <i class="angle tiny double grey right icon">DB.MORE</i>
    </a>
</p>

<div class="ui mini tag label">
    |   :
</div>
</div>
<div class="ui container vertical segment">
    <a href="#">
        <h1 class="ui header">
            | 豆瓣电影
        </h1>
    </a>
    <i class="icon grey small unhide"></i>
    <p>
        介绍的是爬取静态网页，所以请求相应的URL之后，我们需要对其中的信息进行提取，这时候就需要BeautifulSoup库，它可以轻松的找出我们需要的信息
        <a href="http://127.0.0.1:8000/page/db">
            <i class="angle tiny double grey right icon">DB.MORE</i>
        </a>
    </p>

```

```
</p>
<div class="ui mini tag label">
    |   :
</div>
</div>

<div class="ui container vertical segment">
    <a href="#">
        <h1 class="ui header">
            | 天气信息
        </h1>
    </a>
    <i class="icon grey small unhide"></i>
    <p>
        | 动态页面爬取。之所以叫动态页面爬取解析其实是相对于静态下载器与解析器来说的，因为有时候我们使用静态下载器与解析器对一些要爬取的页面进行
        <a href="http://127.0.0.1:8000/page/tq">
            <i class="angle tiny double grey right icon">DB.MORE</i>
        </a>
    </p>
    <div class="ui mini tag label">
        |   :
    </div>
</div>
<div class="ui container vertical segment">
    <a href="#">
        <h1 class="ui header">
            | 京东页面
        </h1>
    </a>
    <i class="icon grey small unhide"></i>
    <p>
        在动态页面爬取的基础上，伪装浏览器，进行爬取。爬虫与反爬虫是一场持续的斗争，在大量爬取的过程中，使用模拟无头浏览器爬取页面信息是一种方
        <a href="http://127.0.0.1:8000/page/jd">
            <i class="angle tiny double grey right icon">DB.MORE</i>
        </a>
    </p>
    <div class="ui mini tag label">
        |   :
    </div>
</div>
<div class="ui container vertical segment">
```

```
|   : )
| </div>
</div>
<div class="ui container vertical segment">
  <a href="#">
    <h1 class="ui header">
      | 淘宝书包
    </h1>
  </a>

  <i class="icon grey small unhide"></i>
  <p>
    使用正则表达式等解析页面信息经行爬取，注：淘宝的robot协议是禁止任何浏览器爬取的。
    <a href="http://127.0.0.1:8000/page/tb">
      | <i class="angle tiny double grey right icon">DB.MORE</i>
    </a>
  </p>

  <div class="ui mini tag label">
    | : )
  </div>
</div>

</div>

{% block body %}{% endblock%}

<div class="ui inverted vertical very padded segment">
  | Mugglecoding®
</div>

{% block foot %}{% endblock%}

</body>
</html>
```

web_db:

```
!DOCTYPE html>
{%
    load staticfiles %
}
<html>
    <head>
        <meta charset="utf-8">
        <title>first web</title>
        <link rel="stylesheet" href="{% static 'css/semantic.css' %}" media="screen" title="no title" charset="utf-8">
        <link href="https://fonts.googleapis.com/css?family=Oswald|Raleway" rel="stylesheet">
    </head>
    <style type="text/css">
        h1 {
            font-family: 'Oswald', sans-serif!important;
            font-size:40px;
        }

        body {
            font-family: 'Raleway', sans-serif;
        }
        p {
            font-family: 'Raleway', sans-serif;
            font-size:18px;
        }
        .ui.vertical.segment.masthead {
            height: 300px;
            background-image: url("{% static 'images/star_banner.jpg' %}");
            background-size: cover;
            background-position: 100% 80%;
        }

        .ui.container.segment {
            width: 800px;
        }

        .ui.center.aligned.header.blogslogon {
            margin-top: 40px;
        }

        .ui.center.aligned.header.blogslogon p {
            margin-top: 10px;
            color: white;
            font-size: 10px;
        }
        .ui.container.nav {
            width: 500px;
        }
    </style>

```

```
        }

    </style>
    {% block css %}{% endblock%}

</head>
<body>
    <div class="ui inverted vertical segment masthead">
        <h1 class="ui center aligned header blogslogon" style="font-size:50px;font-family: 'Raleway', sans-serif!important;">
            跪虫展示
            <p class="ui sub header">
                | 下面为爬取数据展示。
            </p>
        </h1>
    </div>
    <div class="ui container nav">
        <div class="ui borderless text three item menu ">
            <div class="ui simple dropdown item">
                加油
                <i class="dropdown icon"></i>
                <div class="menu">
                    <a class="item" href="">简介</a>
                    <a class="item" href="">简介</a>
                </div>
            </div>
            <a class="item">
                | 努力
            </a>
            <a class="item">
                | 简介
            </a>
        </div>
    </div>
    <div class="ui divider"></div>

    {% block header %}{% endblock%}

    <div class="ui container nav" >
        <br>
        <br>
    </div>
```

```
<br>
<h1>豆瓣电影</h1>
<br>
<br>
<h3><a href='http://127.0.0.1:8000/page/main'>回到页面</a></h3>
<br>
<br>
{% for content in content_list %}

<h3>
    编号: {{content.id}} <br>
    名字: {{content.content}}<br>
</h3>

<br>
<br><br>
{% endfor %}
```

```
{% if data %}

<ul id="pages" class="pagination pagination-sm pagination-xs">
    {% if data.first %}
        <li><a href="?page=1">1</a></li>
    {% endif %}
    {% if data.left %}
        {% if data.left_has_more %}
```

```

    {% if data.left_has_more %}
        <li><span>...</span></li>
    {% endif %}

    {% for i in data.left %}
        <li><a href="?page={{i}}">{{i}}</a></li>
    {% endfor %}

    {% endif %}

    <li class="active"><a href="?page={{data.page}}">{{data.page}}</a></li>

    {% if data.right %}

        {% for i in data.right %}

            <li><a href="?page={{i}}">{{i}}</a></li>
        {% endfor %}

        {% if data.right_has_more %}

            <li><span>...</span></li>
        {% endif %}

        {% endif %}

        {% if data.last %}

            <li><a href="?page={{data.total_pages}}">{{data.total_pages}}</a></li>
        {% endif %}

    </ul>
    {% endif %}
</div>

{% block body %}{% endblock%}

<div class="ui inverted vertical very padded segment">
    </div>

```

```

    {% block body %}{% endblock%}

    <div class="ui inverted vertical very padded segment">
        Mugglecoding®
    </div>

    {% block foot %}{% endblock%}
</body>
</html>

```

web_tq:

```
!DOCTYPE html>
{% load staticfiles %}
<html>
  <head>
    <meta charset="utf-8">
    <title>first web</title>
    <link rel="stylesheet" href="{% static 'css/semantic.css' %}" media="screen" title="no title" charset="utf-8">
    <link href="https://fonts.googleapis.com/css?family=Oswald|Raleway" rel="stylesheet">

    <style type="text/css">
      h1 {
        font-family:'Oswald', sans-serif!important;
        font-size:40px;
      }

      body {
        font-family: 'Raleway', sans-serif;
      }
      p {
        font-family: 'Raleway', sans-serif;
        font-size:18px;
      }
      .ui.vertical.segment.masthead {
        height: 300px;
        background-image: url("{% static 'images/star_banner.jpg' %}");
        background-size: cover;
        background-position: 100% 80%;
      }

      .ui.container.segment {
        width: 800px;
      }

      .ui.center.aligned.header.blogslogon {
        margin-top: 40px;
      }

      .ui.center.aligned.header.blogslogon p {
        margin-top: 10px;
        color: white;
        font-size: 10px;
      }
      .ui.container.nav {
        width: 500px;
      }
    </style>
  {% block css %}{% endblock%}

</head>
<body>
  <div class="ui inverted vertical segment masthead">
    <h1 class="ui center aligned header blogslogon" style="font-size:50px;font-family: 'Raleway', sans-serif!important;">
      跳虫展示
      <p class="ui sub header">
        | 下面为爬取数据展示。
      </p>
    </h1>
  </div>
  <div class="ui container nav">
    <div class="ui borderless text three item menu ">
      <div class="ui simple dropdown item">
        加油
        <i class="dropdown icon"></i>
        <div class="menu">
          <a class="item" href="">简介</a>
          <a class="item" href="">简介</a>
        </div>
      </div>
      <a class="item">
        | 努力
      </a>
      <a class="item">
        | 简介
      </a>
    </div>
  </div>
  <div class="ui divider"></div>
  {% block header %}{% endblock%}

  <div class="ui container nav" >
    <br>
    <br>
  </div>

```

```
  </div>
</body>
```

```
<h1>天气页面</h1> |  
  
<br>  
  
<br>  
<h3><a href='http://127.0.0.1:8000/page/main'>回到页面</a></h3>  
  
<br>  
  
<br>  
  
{% for weather in content_list %}  
    {{weather.id}}&nbsp&nbsp&nbsp&nbsp:&nbsp&nbsp&nbsp&nbsp&nbsp  
    {{weather.wCity}}&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp  
    {{weather.wDate}}&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp  
    {{weather.wWeather}}&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp  
    {{weather.wTemp}}</h3>  
  
<br><br><br>  
  
{% endfor %}  
  
  
{% if data %}  
<ul id="pages" class="pagination pagination-sm pagination-xs">  
    {% if data.first %}  
        <li><a href="?page=1">1</a></li>  
    {% endif %}  
    {% if data.left %}  
        {% if data.left_has_more %}  
            <li><span>...</span></li>  
        {% endif %}  
        {% for i in data.left %}  
            {% if data.left_has_more %}  
                <li><span>...</span></li>  
            {% endif %}  
            <li><a href="?page={{i}}">{{i}}</a></li>  
        {% endfor %}  
        {% endif %}  
        <li class="active"><a href="?page={{data.page}}">{{data.page}}</a></li>  
        {% if data.right %}  
            {% for i in data.right %}  
                <li><a href="?page={{i}}">{{i}}</a></li>  
            {% endfor %}  
            {% if data.right_has_more %}  
                <li><span>...</span></li>  
            {% endif %}  
            {% endif %}  
            {% if data.last %}  
                <li><a href="?page={{data.total_pages}}">{{data.total_pages}}</a></li>  
            {% endif %}  
    {% endif %}  
</ul>  
    {% endif %}  
</div>  
{% block body %}{% endblock%}  
<div class="ui inverted vertical very padded segment">
```

```

        </div>

    {%- block body %}{% endblock%}

    <div class="ui inverted vertical very padded segment">
        |   Mugglecoding
    </div>

    {%- block foot %}{% endblock%}

</body>
</html>

```

Web_jd:

```

!DOCTYPE html>
{% load staticfiles %}
<html>
    <head>
        <meta charset="utf-8">
        <title>first web</title>
        <link rel="stylesheet" href="{% static 'css/semantic.css'%}" media="screen" title="no title" charset="utf-8">
        <link href="https://fonts.googleapis.com/css?family=Oswald|Raleway" rel="stylesheet">

        <style type="text/css">
            h1 {
                font-family:'Oswald', sans-serif!important;
                font-size:40px;
            }

            body {
                font-family: 'Raleway', sans-serif;
            }
            p {
                font-family: 'Raleway', sans-serif;
                font-size:18px;
            }
            .ui.vertical.segment.masthead {
                height: 300px;
                background-image: url("{% static 'images/star_banner.jpg'%}");
                background-size: cover;
                background-position: 100% 80%;
            }

            .ui.container.segment {
                width: 800px;
            }

            .ui.center.aligned.header.blogslogon {
                margin-top: 40px;
            }

            .ui.center.aligned.header.blogslogon p {
                margin-top: 10px;
                color: white;
                font-size: 10px;
            }
            .ui.container.nav {
                width: 500px;
            }
        </style>
    </head>
    <body>
        <div class="ui vertical segment masthead">
            |   Mugglecoding
        </div>

        {%- block foot %}{% endblock%}

    </body>
</html>

```

```
        }

    </style>
    {% block css %}{% endblock%}

</head>
<body>
    <div class="ui inverted vertical segment masthead">
        <h1 class="ui center aligned header blogslogon" style="font-size:50px;font-family: 'Raleway', sans-serif!important;">
            跪虫展示
            <p class="ui sub header">
                | 下面为爬取数据展示。
            </p>
        </h1>
    </div>
    <div class="ui container nav">
        <div class="ui borderless text three item menu ">
            <div class="ui simple dropdown item">
                加油
                <i class="dropdown icon"></i>
                <div class="menu">
                    <a class="item" href="">简介</a>
                    <a class="item" href="">简介</a>
                </div>
            </div>
            <a class="item">
                | 努力
            </a>
            <a class="item">
                | 简介
            </a>
        </div>
    </div>
    <div class="ui divider"></div>
    {% block header %}{% endblock%}
    <div class="ui container nav" >
        <br>
        <br>
    </div>
```

```
<h1>京东手机</h1>
<br>
<br>

<h3><a href='http://127.0.0.1:8000/page/main'>回到页面</a></h3>
<br>
<br>

{% for content in content_list %}
    <h4>
        编号: {{content.mNo}}<br>
        图片: <br>
        <br>
        品牌: {{content.mMark}}<br>
        价格: {{content.mPrice}}<br>
        简介: {{content.mNote}}<br>
    </h4>
    <br><br><br>
    {% endfor %}

{% if data %}
<ul id="pages" class="pagination pagination-sm pagination-xs">
    {% if data.first %}
        <li><a href="?page=1">1</a></li>
    {% endif %}
    {% if data.left %}
        {% if data.left has more %}
```

```

    {% if data.left_has_more %}
        <li><span>...</span></li>
    {% endif %}

    {% for i in data.left %}
        <li><a href="?page={{i}}">{{i}}</a></li>
    {% endfor %}

    {% endif %}

    <li class="active"><a href="?page={{data.page}}">{{data.page}}</a></li>

    {% if data.right %}

        {% for i in data.right %}

            <li><a href="?page={{i}}">{{i}}</a></li>
        {% endfor %}

        {% if data.right_has_more %}

            <li><span>...</span></li>
        {% endif %}

        {% endif %}

        {% if data.last %}

            <li><a href="?page={{data.total_pages}}">{{data.total_pages}}</a></li>
        {% endif %}

    </ul>
    {% endif %}
</div>

{% block body %}{% endblock%}

<div class="ui inverted vertical very padded segment">
    </div>

```

```

    {% block body %}{% endblock%}

    <div class="ui inverted vertical very padded segment">
        Mugglecoding®
    </div>

    {% block foot %}{% endblock%}
</body>
</html>

```

Web_tb:

```
!DOCTYPE html>
{% load staticfiles %}
<html>
  <head>
    <meta charset="utf-8">
    <title>first web</title>
    <link rel="stylesheet" href="{% static 'css/semantic.css' %}" media="screen" title="no title" charset="utf-8">
    <link href="https://fonts.googleapis.com/css?family=Oswald|Raleway" rel="stylesheet">

    <style type="text/css">
      h1 {
        font-family:'Oswald', sans-serif!important;
        font-size:40px;
      }

      body {
        font-family: 'Raleway', sans-serif;
      }
      p {
        font-family: 'Raleway', sans-serif;
        font-size:18px;
      }
      .ui.vertical.segment.masthead {
        height: 300px;
        background-image: url("{% static 'images/star_banner.jpg' %}");
        background-size: cover;
        background-position: 100% 80%;
      }

      .ui.container.segment {
        width: 800px;
      }

      .ui.center.aligned.header.blogslogon {
        margin-top: 40px;
      }

      .ui.center.aligned.header.blogslogon p {
        margin-top: 10px;
        color: white;
        font-size: 10px;
      }
      .ui.container.nav {
        width: 500px;
      }
    </style>
  {% block css %}{% endblock%}
</head>
<body>
```

```
<div class="ui inverted vertical segment masthead">
  <h1 class="ui center aligned header blogslogon" style="font-size:50px;font-family: 'Raleway', sans-serif!important;">
    跳虫展示
    <p class="ui sub header">
      | 下面为爬取数据展示。
    </p>
  </h1>
</div>
<div class="ui container nav">
  <div class="ui borderless text three item menu">
    <div class="ui simple dropdown item">
      加油
      <i class="dropdown icon"></i>
      <div class="menu">
        <a class="item" href="">简介</a>
        <a class="item" href="">简介</a>
      </div>
    </div>
    <a class="item">
      努力
    </a>
    <a class="item">
      简介
    </a>
  </div>
</div>
<div class="ui divider"></div>
  {% block header %}{% endblock%}
<div class="ui container nav" >
  | | <br>
```

```
<div class="ui container nav" >
<br>
<br>
<h1>淘宝书包</h1>
<br>
<br>
<% for shubao in content_list %>
    <h3>
        编号: {{shubao.id}}<br>
        价格: {{shubao.Price}}<br>
        简介: {{shubao.Title}}</h3>
    <br><br>
<% endfor %>
<% if data %>

<ul id="pages" class="pagination pagination-sm pagination-xs">
    <% if data.first %>
        <li><a href="?page=1">1</a></li>
    <% endif %>
    <% if data.left %>
        <% if data.left_has_more %>
            <li><span>...</span></li>
        <% endif %>
        <% for i in data.left %>
            <li><a href="?page={{i}}">{{i}}</a></li>
        <% endfor %>
        <% endif %>
        <li class="active"><a href="?page={{data.page}}">{{data.page}}</a></li>
    <% if data.left_has_more %>
        <li><span>...</span></li>
    <% endif %>
    <% for i in data.left %>
        <li><a href="?page={{i}}">{{i}}</a></li>
    <% endfor %>
    <% endif %>
    <li class="active"><a href="?page={{data.page}}">{{data.page}}</a></li>
    <% if data.right %>
        <% for i in data.right %>
            <li><a href="?page={{i}}">{{i}}</a></li>
        <% endfor %>
        <% endif %>
        <% if data.right_has_more %>
            <li><span>...</span></li>
        <% endif %>
        <% endif %>
        <% if data.last %>
            <li><a href="?page={{data.total_pages}}">{{data.total_pages}}</a></li>
        <% endif %>
    </ul>
<% endif %>
</div>

<% block body %><% endblock %>

<div class="ui inverted vertical very padded segment">
```

```
</div>
{%
  block body
  %}{{% endblock%}

<div class="ui inverted vertical very padded segment">
|   Mugglecoding®
</div>

{%
  block foot
  %}{{% endblock%}

</body>
</html>
```

数据库：

NO	CONTENT
1	肖申克的救赎
2	霸王别姬
3	这个杀手不太冷
4	阿甘正传
5	美丽人生
6	泰坦尼克号
7	千与千寻
8	辛德勒的名单
9	盗梦空间
10	忠犬八公的故事
11	机器人总动员
12	三傻大闹宝莱坞
13	海上钢琴师
14	放牛班的春天
15	楚门的世界
16	大话西游之大圣娶亲
17	星际穿越
18	龙猫
19	教父
20	熔炉
21	无间道
22	疯狂动物城
23	当幸福来敲门
24	怦然心动
25	触不可及
26	蝙蝠侠：黑暗骑士
27	乱世佳人
28	活着
29	少年派的奇幻漂流

左侧树形菜单显示了数据库中的所有表和视图，包括 `information_schema`、`sql`、`performance_schema`、`t`、`t` 表、`auth_group`、`auth_group_permissions`、`auth_permission`、`auth_user`、`auth_user_groups`、`auth_user_user_permissions`、`django_admin_log`、`django_content_type`、`django_migrations`、`django_session`、`firstapp_test`、`showapp_movie`、`showapp_phones`、`showapp_shubao`、`showapp_test1` 和 `showapp_weathers`。

wCity	wDate	wWeather	wTemp
▶ 上海	1日 (周三)	小雨转多云	23°C/17°C
上海	26日 (明天)	多云转晴	18°C/12°C
上海	27日 (后天)	多云	18°C/14°C
上海	28日 (周日)	小雨转多云	23°C/17°C
上海	29日 (周一)	中雨转阴	24°C/17°C
上海	30日 (周二)	阴转小雨	22°C/17°C
北京	1日 (周三)	多云	24°C/14°C
北京	26日 (明天)	晴转多云	21°C/10°C
北京	27日 (后天)	多云	18°C/7°C
北京	28日 (周日)	多云	21°C/9°C
北京	29日 (周一)	多云转小雨	24°C/13°C
北京	30日 (周二)	多云	26°C/14°C
广州	1日 (周三)	大到暴雨转多云	26°C/20°C
广州	26日 (明天)	中雨转中到大雨	28°C/24°C
广州	27日 (后天)	中到大雨转雷阵雨	28°C/24°C
广州	28日 (周日)	雷阵雨	29°C/25°C
广州	29日 (周一)	雷阵雨转中雨	30°C/25°C
广州	30日 (周二)	中雨转大到暴雨	30°C/22°C
深圳	1日 (周三)	暴雨转阵雨	27°C/23°C
深圳	26日 (明天)	中雨转大雨	29°C/24°C
深圳	27日 (后天)	大雨转雷阵雨	27°C/23°C
深圳	28日 (周日)	雷阵雨	28°C/24°C
深圳	29日 (周一)	雷阵雨	30°C/25°C
深圳	30日 (周二)	雷阵雨转暴雨	30°C/23°C

id	Price	Title
1	45.80	小学生书包男生1-3-4-6年级6-12周岁儿童
2	39.90	迪卡侬双肩包运动背包男女健身包书包儿童学生户外旅行包KIPSTA
3	119.00	kk树书包小学生女孩6-12周岁儿童1-3-6年级女童双肩背包护脊减负
4	499.00	Fjallraven/北极狐双肩包kanken classic书包女户外旅行背包23510
5	129.00	小米双肩包简约休闲多功能书包男女笔记本电脑包时尚潮流旅行背包
6	258.00	电视剧款JanSport旗舰店官网杰斯伯双肩包时尚女书包背包男大容量
7	348.00	爆款anello官方旗舰店日本ins潮风双肩女背包男离家出走包书包
8	199.00	小米 米兔儿童书包 6-12岁男女小学生潮双肩背包幼儿园大容量背包
9	79.00	双肩包男士背包大容量旅行包电脑休闲女时尚潮流高中初中学生书包
10	109.00	七匹狼商务双肩包男书包中学生女电脑包旅行包休闲男士背包大容量
11	148.00	佑一良品男士背包双肩包男韩版大学生书包男时尚潮流大容量旅行包
▶	12 69.00	巴布豆旗舰店书包1-3年级护脊减负儿童书包男4-6小学生书包轻便
13	299.00	BOPAI博牌电脑背包男户外旅行休闲双肩包商务书包出差多功能男包
14	49.00	小学生书包6-12周岁 女儿童双肩包 3-5年级女童背包 1-3年级女孩
15	45.80	儿童书包小学生男童1-3年级6-12周岁4-6年级男孩双肩背包轻便减负
16	59.80	商务背包男士双肩包韩版潮流旅行包休闲女学生书包简约时尚电脑包
17	168.00	双肩包男书包男士时尚潮流青年休闲简约潮牌旅行背包大学生电脑包
18	69.00	迪士尼书包小学生男女1-3-4-6年级米奇减负背包儿童书包8-10-12岁
19	119.00	巴朗商务双肩包休闲时尚潮流大学生书包15.6寸电脑包男士背包男潮
20	99.00	米熙休闲运动背包双肩包女书包中学生男韩版时尚大容量旅游旅行包
21	195.02	国家地理背包女运动户外时尚双肩包男牛津布旅行防水学生情侣书包
22	79.00	双肩包女士2019新款小韩版百搭时尚书包2019旅行防盗牛津帆布背包
23	89.00	迪士尼书包小学生男童1-3-4五年级6-12周岁女孩儿童减负双肩背包
24	69.00	双肩包女2019新款潮牌韩版时尚百搭女士休闲pu软皮小背包旅行书包
25	138.00	瑞士军刀双肩包男 背包休闲商务旅行大容量瑞士书包电脑男士户外
26	139.00	花花公子男士双肩包时尚潮流休闲电脑旅行书包潮牌大学生帆布背包
27	89.00	HK双肩包男简约个性书包韩版时尚潮流休闲电脑包户外旅行轻便背包
28	49.00	背包男双肩包休闲大容量旅行包时尚潮流韩版高中生初中学生书包男
29	69.00	休闲双肩包男士韩版简约电脑旅行背包女时尚潮流初中高中学生书包

id	mNo	mMark	mPrice	mNote	mFile
1	000001	OPPO	3599.00	OPPO Reno 全面屏拍照手机	000001.jpg
2	000002	Apple	5899.00	Apple iPhone XR (A2108)	000002.jpg
3	000003	【KPL官方比赛用机】vivo	3298.00	【KPL官方比赛用机】vivo i	000003.jpg
4	000004	荣耀8X	1299.00	荣耀8X 千元屏霸 91%屏占比	000004.jpg
5	000005	荣耀10青春版	1299.00	荣耀10青春版 幻彩渐变 24	000005.jpg
6	000006	vivo	799.00	vivo U1 水滴全面屏 AI智慧	000006.jpg
7	000007	荣耀V20	2799.00	荣耀V20 胡歌同款 麒麟980	000007.jpg
8	000008	vivo	3598.00	vivo X27 8GB+256GB大内	000008.jpg
9	000009	OPPO	2999.00	OPPO Reno手机 新品 全面	000009.jpg
10	000010	小米	1199.00	小米 红米Redmi Note7 幻	000010.jpg
11	000011	荣耀畅玩8C两天一充	899.00	荣耀畅玩8C两天一充 莱茵护	000011.jpg
12	000012	小米	799.00	小米 红米6 4GB+64GB 铂	000012.jpg
13	000013	黑鲨游戏手机2	3499.00	黑鲨游戏手机2 8GB+128G	000013.jpg
14	000014	Apple	6199.00	Apple iPhone X (A1865)	6 000014.jpg
15	000015	小米	799.00	小米 红米Redmi 7 幻彩渐变	000015.jpg
16	000016	小米8SE	1599.00	小米8SE 全面屏智能游戏拍	000016.jpg
17	000017	三星	6999.00	三星 Galaxy S10+ 8GB+12	000017.jpg
18	000018	Apple	9699.00	Apple iPhone XS Max (A2	000018.jpg
19	000019	Apple	3799.00	Apple iPhone 8 (A1863)	6 000019.jpg
20	000020	vivo	1598.00	vivo Z3 6GB+64GB 极光蓝	000020.jpg
21	000021	华为	3999.00	华为 HUAWEI Mate 20 麒	000021.jpg
22	000022	vivo	2298.00	vivo S1 6GB+128GB 宠爱	000022.jpg
23	000023	Apple	4699.00	Apple iPhone 8 Plus (A18	000023.jpg
24	000024	小米8青春版	1499.00	小米8青春版 镜面渐变AI双摄	000024.jpg
25	000025	三星	1448.00	三星 Galaxy A6s 6GB+64G	000025.jpg
26	000026	荣耀10	2199.00	荣耀10 GT游戏加速 AIS手	000026.jpg
27	000027	诺基亚	1099.00	诺基亚 NOKIA X6 6GB+64	000027.jpg
28	000028	华为	1999.00	华为 HUAWEI nova 4e 32	000028.jpg

补充:

多进程:

```

#!/usr/bin/env python
# -*- coding=utf-8 -*-

from multiprocessing import Process, Queue

import time
from lxml import etree
import requests

class DouBanSpider(Process):
    def __init__(self, url, q):
        # 重写父类的__init__方法
        super(DouBanSpider, self).__init__()
        self.url = url
        self.q = q
        self.headers = {
            'Host': 'movie.douban.com',
            'Referer': 'https://movie.douban.com/top250?start=225&filter=',
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.104 Safari/537.36',
        }

    def run(self):
        self.parse_page()

    def send_request(self, url):
        """
        用来发送请求的方法
        :return: 返回网页源码
        """

        # 请求出错时，重复请求3次,
        i = 0
        while i <= 3:
            try:
                print(u'[INFO]请求url:' + url)
                return requests.get(url=url, headers=self.headers).content
            except Exception as e:
                print(u'[INFO] %s' % (e, url))
                i += 1

    def parse_page(self):
        """
        解析网站源码，并采用xpath提取 电影名称和平分放到队列中
        :return:
        """
        response = self.send_request(self.url)

        response = self.send_request(self.url)
        html = etree.HTML(response)
        # 抓取到一页的电影数据
        node_list = html.xpath("//div[@class='info']")
        for move in node_list:
            # 电影名称
            title = move.xpath('.//a/span/text()')[0]
            # 评分
            score = move.xpath('.//div[@class="bd"]/span[@class="rating_num"]/text()')[0]

            # 将每一部电影的名称跟评分加入队列
            self.q.put(score + "\t" + title)

    def main():
        # 创建一个队列用来保存进程读取到的数据
        q = Queue()
        base_url = 'https://movie.douban.com/top250?start='
        # 构造所有url
        url_list = [base_url+str(num) for num in range(0, 225+1, 25)]

        # 保存进程
        Process_list = []
        # 创建并启动进程
        for url in url_list:
            p = DouBanSpider(url, q)
            p.start()
            Process_list.append(p)

        # 让主线程等待子进程执行完成
        for i in Process_list:
            i.join()

        while not q.empty():
            print(q.get())

    if __name__ == "__main__":
        start = time.time()
        main()
        print('[info]耗时: %s' % (time.time() - start))

```

多线程：

```

#!/usr/bin/env python
# -*- coding=utf-8 -*-

from threading import Thread
from queue import Queue
import time
from lxml import etree
import requests

class DoubanSpider(Thread):
    def __init__(self,url,q):
        super(DoubanSpider,self).__init__()
        self.url=url
        self.q=q
        self.headers={
            'Cookie': 'll="118282"; bid=ctyiEarSLfw; ps=y; __yadk_uid=0Sr85yZ9d4bEeLKhv4w36950FOPoedzC; dbcl2="155150959:0Eu4dds1Gio"; as="https://movie.douban.com";',
            'Host': 'movie.douban.com',
            'Referer': 'https://movie.douban.com/top250?start=225&filter=',
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.104 Safari/537.36',
        }
    def run(self):
        self.parse_page()

    def send_request(self,url):
        ...
        # 用来发送请求的方法
        :return: 返回网页源码
        ...
        # 请求出错时，重试请求3次,
        i = 0
        while i <= 3:
            try:
                print (u"[INFO]请求url:"+url)
                html = requests.get(url=url,headers=self.headers).content
            except Exception as e:
                print (u'[INFO] %s%s' % (e,url))
                i += 1
            else:
                return html

    def parse_page(self):
        ...
        # 解析网站源码，并采用xpath提取 电影名称和平分放到队列中
        :return:
        ...
        response = self.send_request(self.url)

```

```

        response = self.send_request(self.url)
        html = etree.HTML(response)
        # 获取到一页的电影数据
        node_list = html.xpath("//div[@class='info']")
        for move in node_list:
            # 电影名称
            title = move.xpath('.//a/span/text()')[0]
            # 评分
            score = move.xpath('.//div[@class="bd"]//span[@class="rating_num"]/text()')[0]

            # 将每一部电影的名称跟评分加入到队列
            self.q.put(score + "\t" + title)

    def main():
        # 创建一个队列用来保存进程获取到的数据
        q = Queue()
        base_url = 'https://movie.douban.com/top250?start='
        # 构造所有url
        url_list = [base_url+str(num) for num in range(0,225+1,25)]

        # 保存线程
        Thread_list = []
        # 创建并启动线程
        for url in url_list:
            p = DoubanSpider(url,q)
            p.start()
            Thread_list.append(p)

        # 让主线程等待子线程执行完成
        for i in Thread_list:
            i.join()

        while not q.empty():
            print (q.get())

    if __name__=="__main__":
        start = time.time()
        main()
        print ('[info]耗时: %s'%(time.time()-start))

```

Scrapy: (缓存) 数据库

```
urls.py ...\\showapp urls.py ...\\showpage settings.py wsgi.py views.py down_img.py slave.py x douban.py jingdong.py taobao.py tianqi.py

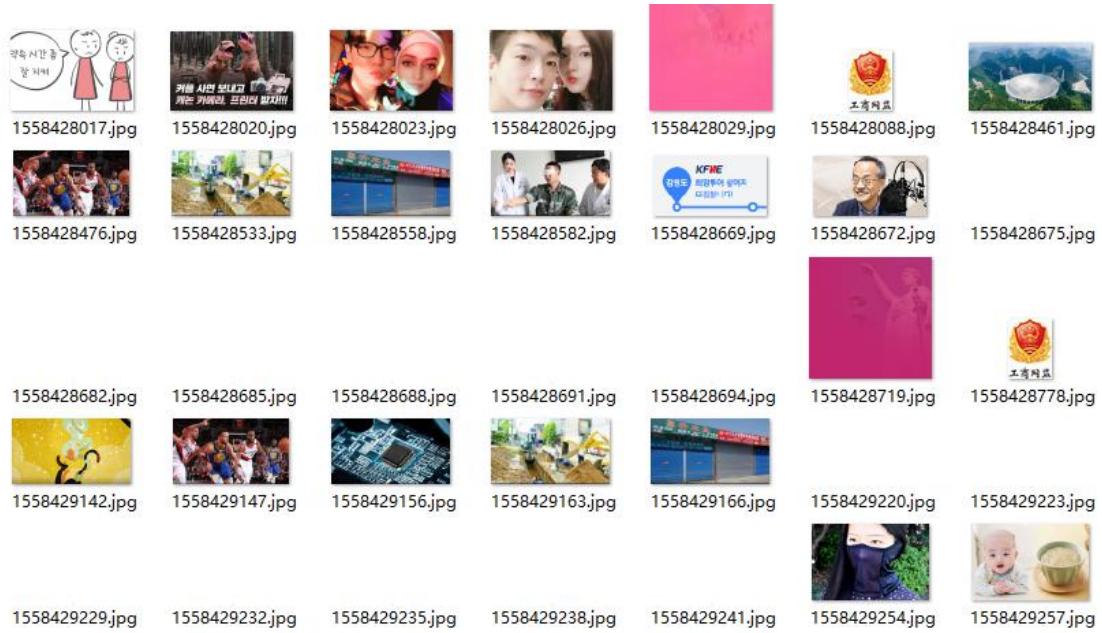
1 import requests
2 from bs4 import BeautifulSoup
3 import re
4 import time
5 from redis import Redis
6 headers={ 'User-Agent':'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36' }
7
8 def push_redis_list():
9     r = Redis(host='127.0.0.1', port=7799 ,password='')
10    print (r.keys('*'))
11
12    link_list = []
13    with open('alexa.txt', 'r') as file:
14        file_list = file.readlines()
15        for eachone in file_list:
16            link = eachone.split('\\t')[1]
17            link = link.replace('\\n', '')
18            link_list.append(link)
19            if len(link_list) == 100:
20                break
21
22    for url in link_list:
23        response = requests.get(url, headers=headers, timeout=20)
24        soup = BeautifulSoup(response.text, 'lxml')
25        img_list = soup.findall('img')
26        for img in img_list:
27            img_url = img['src']
28            if img_url != '':
29                print ('加入的图片url: ', img_url)
30                r.lpush('img_url',img_url)
31        print ('现在图片链接的个数为', r.llen('img_url'))
32    return
33
34 def get_img():
35     r = Redis(host='127.0.0.1', port=7799 ,password='')
36     while True:
37         try:
38             url = r.lpop('img_url')
39             url = url.decode('ascii')
40             if url[:2] == '//':
41                 url = 'http:' + url
42             print (url)
43         try:
44             response = requests.get(url, headers=headers,timeout = 20)
45             name = int(time.time())
46             f = open(str(name)+ url[-4:], 'wb')
47             f.write(response.content)
48             f.close()
49             print ('已经获取图片', url)
50         except Exception as e:
51             print ('已经获取图片', url)
52         except Exception as e:
53             print ('[!]', '获取图片过程出问题', e)
54             time.sleep(3)
55         except Exception as e:
56             print ('e')
57             time.sleep(10)
58             break
59     return
60
61 if __name__ == '__main__':
62     this_machine = 'slave'
63     print ('开始分布式爬虫')
64     if this_machine == 'master':
65         push_redis_list()
66     else:
67         get_img()
```

```

urls.py ...\\showapp      urls.py ...\\showpage      settings.py      wsgi.py      views.py      down_img.py x  douban.py      jingdong.py      taobao.py      tia
1 import requests
2 from bs4 import BeautifulSoup
3 import re
4 import time
5 from redis import Redis
6 headers={ 'User-Agent':'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36' }
7
8 def push_redis_list():
9     r = Redis(host='127.0.0.1', port=7799 ,password='')
0     print (r.keys('*'))
1
2     link_list = []
3     with open('alexa.txt', 'r') as file:
4         file_list = file.readlines()
5         for eachone in file_list:
6             link = eachone.split('\\t')[1]
7             link = link.replace('\\n','')
8             link_list.append(link)
9             if len(link_list) == 100:
0                 break
1
2     for url in link_list:
3         response = requests.get(url, headers=headers, timeout=20)
4         soup = BeautifulSoup(response.text, 'lxml')
5         img_list = soup.findAll('img')
6         for img in img_list:
7             img_url = img['src']
8             if img_url != '':
9                 print ("加入的图片url: ", img_url)
0                 r.rpush('img_url',img_url)
1             print ('现在图片链接的个数为', r.llen('img_url'))
2
3     return
4
5 def get_img():
6     r = Redis(host='127.0.0.1', port=7799 ,password='')
7     while True:
8         try:
9             url = r.lpop('img_url')
0             url = url.decode('ascii')
1             try:
2                 response = requests.get(url, headers=headers,timeout = 4)
3                 name = int(time.time())
4                 f = open(str(name)+ url[-4:], 'wb')
5                 f.write(response.content)
6                 f.close()
7                 print ('已经获取图片', url)
8             except Exception as e:
9                 print ('爬取图片过程中出问题', e)
0                 time.sleep(3)
1             except Exception as e:
2                 time.sleep(3)
3                 print (e)
4                 time.sleep(10)
5                 break
6             return
7
8 if __name__ == '__main__':
9     this_machine = 'master'
0     print ('开始分布式爬虫')
1     if this_machine == 'master':
2         push_redis_list()
3     else:
4         get_img()

```

图片：



12306 登陆验证：

- 1: 手动验证/半自动验证
- 2: session cookie 验证
- 3: ocr 验证

```
12500_0.py 12500_1.py 12500_2.py 12500_3.py 12500_4.py 12500_5.py 12500_6.py 12500_7.py 12500_8.py 12500_9.py
1 import requests
2 import config
3
4 person=requests.session()
5 login_url='https://kyfw.12306.cn/otn/login/init'
6
7 login_response=person.get(login_url)
8
9 captcha_url='https://kyfw.12306.cn/passport/captcha/captcha-image?login_site=E&module=login&rand=sjrand&0.785280601210562'
0
1 captcha_response=person.get(captcha_url)
2 captcha_content=captcha_response.content
3
4 fb=open('captcha.jpg','wb')
5 fb.write(captcha_content)
6 fb.close
7
8 check_url='https://kyfw.12306.cn/passport/captcha/captcha-check'
9 data={
10     'answer':input('请输入验证码坐标》》》:'),
11     'login_site':'E',
12     'rand':'sjrand'
13 }
14 check_response=person.post(check_url,data=data)
15 res=check_response.json()
16 if not res['result_code']=='4':
17     | exit('验证码校验失败')
18
19 login_url='https://kyfw.12306.cn/passport/web/login'
20 login_data={
21     'username': config.username,
22     'password': config.password,
23     'appid':'otn'
24 }
25 login_res=person.post(login_url,data=login_url)
26 if login_res.json()['result_code']!=0:
27     | exit('用户名密码错误')
28
29 token_url='https://kyfw.12306.cn/passport/web/auth/uamtk'
30 token_data={
31     'appid':'otn',
32 }
33 token_response=person.post(token_url,data=token_data)
34 token_res=token_response.json()
35
36 auth_url='https://kyfw.12306.cn/otn/uamauthclient'
37 auth_data={
38     'tk':token_res['newapptk']
39 }
40
41 auth_response=person.post(auth_url,data=auth_data)
42 print(auth_response.text)
```

```

12306_登陆.py ● 12306_login.py 12306_lo.py 12306_2.py 12306_people.py 12306_selenium.py 12306_ocr.py

1 import re
2 import requests
3 import base64
4 import time
5 import os
6
7
8 class check_login():
9     def getImg(self):
10         headers = {'User-Agent':
11             'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'
12         }
13         Xq_url = 'https://kyfw.12306.cn/passport/captcha/captcha-image64' #生成验证码的url
14         Xq_parms = {
15             "login_site": "E",
16             "module": "login",
17             "rand": "sjrand",
18             "15669770297": "",
19             "callback": "jQuery19109087628126888729_15669770297",
20             "_": "15669770297",
21         }
22         sess=requests.session()
23         response=sess.get(url=Xq_url,params=Xq_parms,headers=headers).text
24         image_b64=re.findall('image": "(.*?)"',response)[0]
25         image=base64.b64decode(image_b64)
26         with open('image.jpg','wb') as f:
27             f.write(image)
28         self.sess=sess
29
30     def check_result(self):
31         Xq_url="http://littlebigluo.qicp.net:47720/"
32         sess=requests.session()
33         response1=sess.post(url=Xq_url,data={"type":"1"},files={'pic_xxfile':open('YZ_image.jpg','rb')})
34         result=[]
35         try:
36             for i in re.findall('<B>(.*?)</B>',response1.text)[0].split(" "):
37                 result.append(int(i))
38         except:
39             print("该验证码网站繁忙")
40         coord_data = [
41             "1": "40,40", "2": "120,40", "3": "180,40", "4": "250,40", "5": "40,100", "6": "120,100", "7": "180,100", "8": "250,100",
42         ]
43         answerlist = []
44         print('选中图片为:',result)
45         for i in result:
46             answerlist.append(coord_data[str(i)])
47         print('坐标为: ' + ','.join(answerlist))
48         answer = ','.join(answerlist)
49         self.answer=answer
50
51
52     def login(self):
53         check_url="https://kyfw.12306.cn/passport/captcha/captcha-check" #验证验证码的url
54         Sy_url="https://kyfw.12306.cn/passport/web/login" #登录的url
55
56         check_headers = {
57             'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131 Safari/537.36',
58         }
59
60         Sy_headers = {
61             'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131 Safari/537.36',
62             'Accept-Encoding': 'gzip, deflate, br',
63             'Accept-Language': 'zh-CN,zh;q=0.9',
64             'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8',
65             'Origin': 'https://kyfw.12306.cn',
66             'Referer': 'https://kyfw.12306.cn/otn/resources/login.html',
67             'Accept': 'application/json, text/javascript, */*; q=0.01',
68         }
69
70
71         username = input('请输入用户名: ')
72         password = input('请输入密码: ')
73
74         log_data = {
75             "username": username,
76             "password": password,
77             "appid": "otn",
78             "#answer": self.answer,
79         }
80
81         #YZ_url
82         log_parms = {
83             "callback": "jQuery19105010300528763358_1559733968819",
84             "answer": self.answer,
85             "rand": "sjrand",
86             "login_site": "E",
87             "_": "15669770297",
88         }
89
90         #发送图片验证码请求
91         response2=self.sess.get(url=check_url,params=log_parms,headers=check_headers).text
92         #获得图片验证码信息
93         print(re.findall('{"result_message": "(.*?)"',response2))
94
95         #增加cookies
96         self.sess.cookies.update([
97             'RAIL_EXPIRATION': '15669770297',
98             'RAIL_DEVICEID': 'p3xGaEygCtCN3Q1YYwImAt9vwLj1LB4oZltVN07EBp_UARzbREL1Hz1Vd2B1_HL980K1Pl__tibGwfXfIt6zbappKNeFwxcbQ-BXZ1dxgsQ!V LJBS_IR6k13aAaoH2A99dz55yCS97eNTJTxmNsAqXihoK0p'
99         })

```

```

8     response3=self.sess.post(url=Sy_url,data=log_data,headers=Sy_headers)
9     #追回编码后的数据
10    response3.encoding='utf-8'
11    print(re.findall("result_message": "(.*?)" ,response3.text))
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```

```

12306_登陆.py ● 12306_login.py 12306_lo.py 12306_2.py 12306_people.py 12306_selenium.py ✘ 12306_ocr.py
1  from selenium import webdriver
2  from selenium.webdriver.common.keys import Keys
3  from selenium.webdriver.common.by import By
4  from selenium.webdriver.support.ui import WebDriverWait
5  from selenium.webdriver.support import expected_conditions as EC
6  from selenium.webdriver.common.action_chains import ActionChains
7  import requests
8  import base64
9  import re
10 import time
11
12
13 class otc():
14
15     def __init__(self):
16         self.coordinate = [[-105, -20], [-35, -20], [40, -20], [110, -20], [-105, 50], [-35, 50], [40, 50], [110, 50]]
17
18     def login(self):
19         login_url = "https://kyfw.12306.cn/otn/resources/login.html"
20         driver = webdriver.Chrome()
21         driver.set_window_size(1200, 900)
22         driver.get(login_url)
23         account = driver.find_element_by_class_name("login-hd-account")
24         account.click()
25         userName = driver.find_element_by_id("J-userName")
26         userName.send_keys("*****")
27         password = driver.find_element_by_id("J-password")
28         password.send_keys("*****")
29         self.driver = driver
30
31
32
33     def getVerifyImage(self):
34         try:
35             img_element = WebDriverWait(self.driver, 100).until(
36                 EC.presence_of_element_located((By.ID, "J-loginImg")))
37         except Exception as ex:
38             print("网络繁忙, 请稍后尝试")
39             base64_str = img_element.get_attribute("src").split(",")[-1]
40             imgdata = base64.b64decode(base64_str)
41             with open('verify.jpg', 'wb') as file:
42                 file.write(imgdata)
43             self.img_element = img_element
44
45
46
47     def getVerifyResult(self):
48         url = "http://littlebigluo.qicp.net:47720/"
49         response = requests.request(
50             "POST",

```

```
06_登陆.py • 12306_login.py 12306_lo.py 12306_2.py 12306_people.py 12306_selenium.py ✘ 12306_ocr.py

def getVerifyResult(self):
    url = "http://littlebigluo.qicp.net:47720/"
    response = requests.request(
        "POST",
        url,
        data={"type": "1"},
        files={'pic_xxfile': open('verify.jpg', 'rb')})
    result = []
    print(response.text)
    for i in re.findall("<B>(.*)</B>", response.text)[0].split(" "):
        result.append(int(i) - 1)
    self.result = result
    print(result)

def moveAndClick(self):
    try:
        Action = ActionChains(self.driver)
        for i in self.result:
            Action.move_to_element(self.img_element).move_by_offset(self.coordinate[i][0], self.coordinate[i][1]).click()
        Action.perform()
    except Exception as ex:
        print(ex.message())

def submit(self):
    self.driver.find_element_by_id("J-login").click()

def __call__(self):
    self.login()
    time.sleep(3)
    self.getVerifyImage()
    time.sleep(1)
    self.getVerifyResult()
    time.sleep(1)
    self.moveAndClick()
    time.sleep(1)
    self.submit()
    time.sleep(10000)

otc()()
```

总结：

一：

1. 正确爬取爬虫代码，并将其存储在数据库中。
2. 使用 django，设计首页，包括样式，链接跳转，子页面。
3. 子页面中{{classname: idname}}显示数据库调取的信息。
4. 其中，我使用了 semantic ui 框架来作为我页面的美化。
5. 但是在分页面中我是用{% extends 'xxx.html'%}来继承主页面的样式时由于版本的原因，不能经行覆盖。
6. 代码上传，编写报告，完成。

first web 127.0.0.1:8000/page/main/

爬虫展示

下图为爬虫数据展示。

介绍信息

Name: 李雪萌
Num: 16219111337
class: 16 计算机科学与技术三班
Project: include 爬取豆瓣电影、天气预报、京东商城、淘宝书包
CDB:MORE

豆瓣电影

介绍的是爬取静态网页，所以请求相应的URL之后，我们需要对其中的信息进行提取，这时候就需要BeautifulSoup库，它可以轻松的找出我们需要的信息，当然，有时候借助正则表达式会更快地帮助我们抽取网页中我们需要的信息。 CDB:MORE

天气信息

动态页面爬取，之所以叫动态页面爬取解析其实是相对于静态下载器与解析器来说的，因为有时候我们使用静态下载器与解析器对一些要爬取的页面进行解析时竟然没有任何数据，其实大多原因都是我们要爬取的元素是JS动态生成的，譬如我们爬取页面，你会发现随着我们手指上滑其页面会无限制的上拉加载更多，也就是常说的数据流，这时候我们就会觉得该系列前面介绍的爬取方式似乎完全无能为力了，所以我们需要寻求新的爬取解析方式，也就是动态页面爬取解析，其流行的核心主流思路是动态页面逆向分析爬取和模拟浏览器行为爬取。 CDB:MORE

豆瓣电影

介绍的是爬取静态网页，所以请求相应的URL之后，我们需要对其中的信息进行提取，这时候就需要BeautifulSoup库，它可以轻松的找出我们需要的信息，当然，有时候借助正则表达式会更快地帮助我们抽取网页中我们需要的信息。 CDB:MORE

天气信息

动态页面爬取，之所以叫动态页面爬取解析其实是相对于静态下载器与解析器来说的，因为有时候我们使用静态下载器与解析器对一些要爬取的页面进行解析时竟然没有任何数据，其实大多原因都是我们要爬取的元素是JS动态生成的，譬如我们爬取页面，你会发现随着我们手指上滑其页面会无限制的上拉加载更多，也就是常说的数据流，这时候我们就会觉得该系列前面介绍的爬取方式似乎完全无能为力了，所以我们需要寻求新的爬取解析方式，也就是动态页面爬取解析，其流行的核心主流思路是动态页面逆向分析爬取和模拟浏览器行为爬取。 CDB:MORE

京东页面

在动态页面爬取的基础上，伪装浏览器，进行爬取。爬虫与反爬虫是一场持续的斗争，在大量爬取的过程中，使用模拟无头浏览器爬取页面信息是一种方便的方式。 CDB:MORE

淘宝书包

使用正则表达式等解析页面信息进行爬取，注：淘宝的robot协议是禁止任何浏览器爬取的。 CDB:MORE

MuggleCoding®

豆瓣电影

回到页面

TOP250电影数据

排名	电影名
1	肖申克的救赎
2	霸王别姬
3	这个杀手不太冷
4	阿甘正传
5	美丽人生
6	泰坦尼克号
7	肖申克的救赎
8	泰坦尼克号
9	盗梦空间
10	忠犬八公的故事

Muggiecoding®

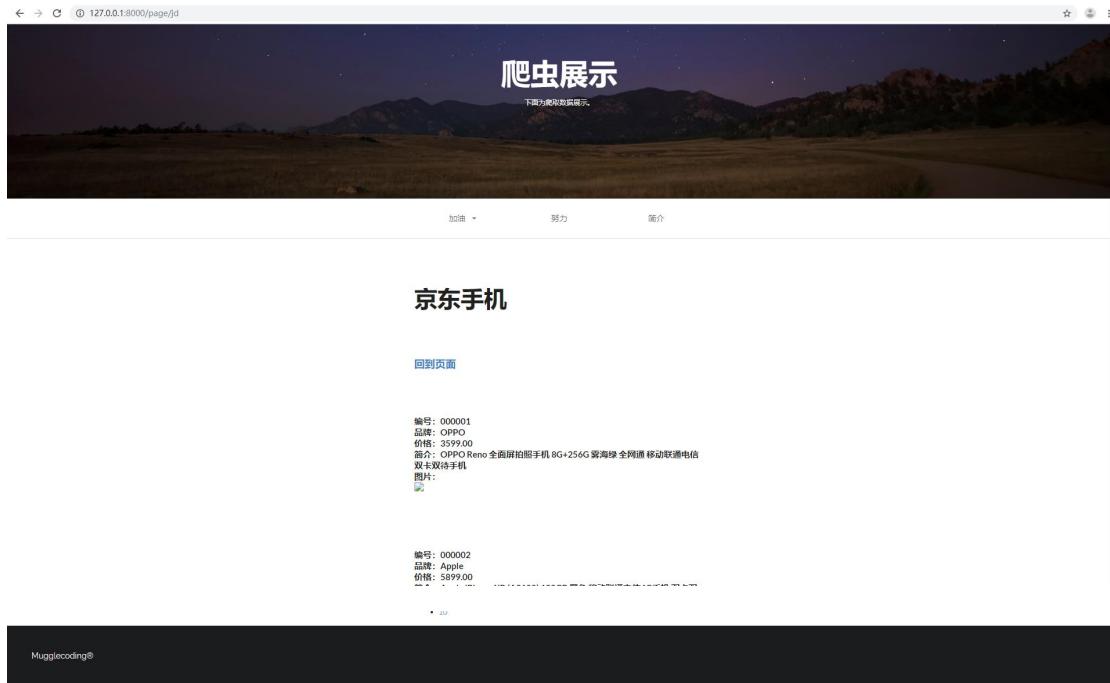
天气页面

回到页面

上海
1日 (周三)
小雨转多云
23°C/17°C

上海
26日 (明天)
多云转晴
18°C/12°C

Muggiecoding®



二：

Django 数据展示有两种方式：一种是直接读取爬虫存储到数据库的数据；还有一种是 django 项目内存储数据并显示

三：

验证码登录，12306全自动登录

搜索引擎的实现：

- 1.从互联网抓取网页
- 2.建立索引数据库
- 3.在索引数据库中搜索
- 4.对搜索结果进行处理排序

