



《爬虫》设计

|  |  |
| --- | --- |
| 题 目： | 爬虫期末大作业 |
| 二级学院： | 数学与信息工程学院 |
| 班 级： | 16计算机科学与技术三 |
| 姓 名： | 童骋 |
| 学 号： | 14219110112 |
| 完成日期： | 2019年6月17日 |

温州大学瓯江学院教务部

二○一二年十一月制

**爬虫项目罗列：**

**（一）爬虫期中作业：**

1. **豆瓣top250上获取电影名称的数据**
2. **爬取天气预报数据**
3. **爬取淘宝书包数据**
4. **爬取京东手机数据**

**（二）爬虫期末作业：**

**1. 通过selenium实现 12306自动登陆**

实现步骤：1.从页面获取到验证码图片：

2. 然后通过上传图片查找出正确的物品

3.实现登录跳转

**1.豆瓣top250上获取电影名称的数据**

**代码：**

import requests

from lxml import etree

import urllib.request

import pymysql

from bs4 import BeautifulSoup

conn=pymysql.connect(host='localhost',user='root',passwd='1234',db='test',charset="utf8")

cursor=conn.cursor()

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'}

for i in range(0,10):

url = 'https://movie.douban.com/top250?start='+str(25\*i)

r = requests.get(url,headers=headers)

html = etree.HTML(r.text)

datas=html.xpath('//ol[@class="grid\_view"]/li')

a=0

for data in datas:

title=data.xpath('div/div[2]/div[@class="hd"]/a/span[1]/text()')

img=data.xpath('div/div[1]/a/img/@src')

urllib.request.urlretrieve(img[0],filename="G:top250/"+str(i\*25+a+1)+".jpg")

a+=1

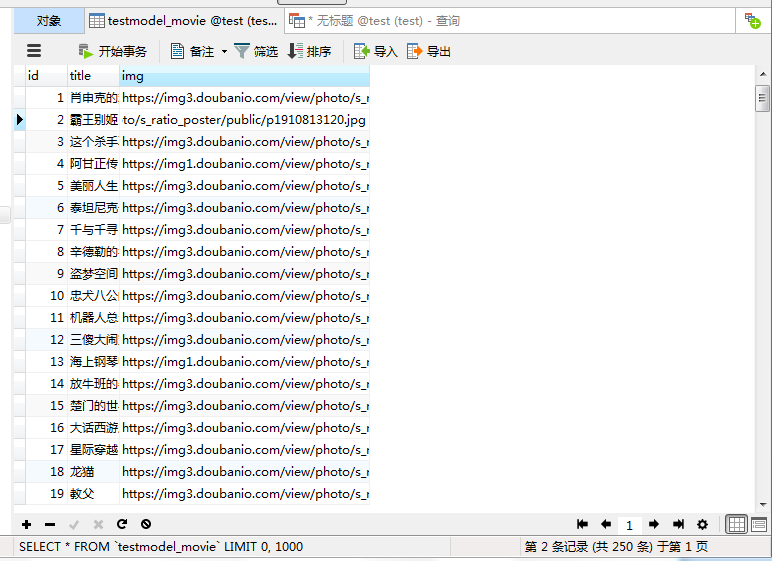
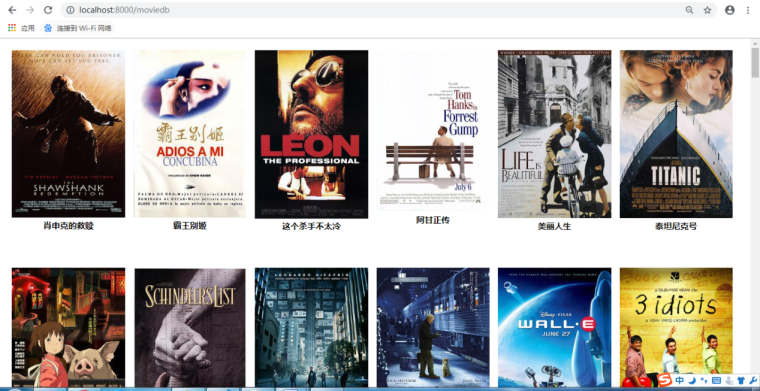
cursor.execute("insert into testmodel\_movie(title,img) values(%s,%s)",(title,img))

cursor.close()

conn.commit()

conn.close()

**数据存储在SQL中： Django界面显示截图：**



通过Django将数据显示在网页上：

项目二：静态爬取5个城市一周内的天气数据

**2.爬取天气预报数据**

代码：

from bs4 import BeautifulSoup

from bs4 import UnicodeDammit

import urllib.request

import pymysql

conn=pymysql.connect(host='localhost',user='root',passwd='1234',db='test',charset="utf8")

cursor=conn.cursor()

headers={'user-agent':'Mozilla/5.0(Windows;U;Windows NT 6.0 x64;en-us;rv:1.9pre)Gecko/2008072421 MineField/3.0.2pre'}

citycode={"北京":"101010100","上海":"101020100","广州":"101280101","深圳":"101280601"}

for city in citycode:

url="http://www.weather.com.cn/weather/"+citycode[city]+".shtml"

try:

req=urllib.request.Request(url,headers=headers)

data=urllib.request.urlopen(req)

data=data.read()

dammint=UnicodeDammit(data,["utf-8","gbk"])

data=dammint.unicode\_markup

soup=BeautifulSoup(data,"lxml")

lis=soup.select("ul[class='t clearfix'] li")

n=0

for li in lis:

try:

date=li.select('h1')[0].text

print(date)

weather=li.select("p[class='wea']")[0].text

if n>0:

temp=li.select("p[class='tem'] span")[0].text+"/"+li.select("p[class='tem'] i")[0].text

else:

temp=li.select("p[class='tem'] i")[0].text

cursor.execute("insert into testmodel\_weather(city,date,weather,temp) values(%s,%s,%s,%s)",(city,date,weather,temp))

n=n+1

except Exception as err:

print(err)

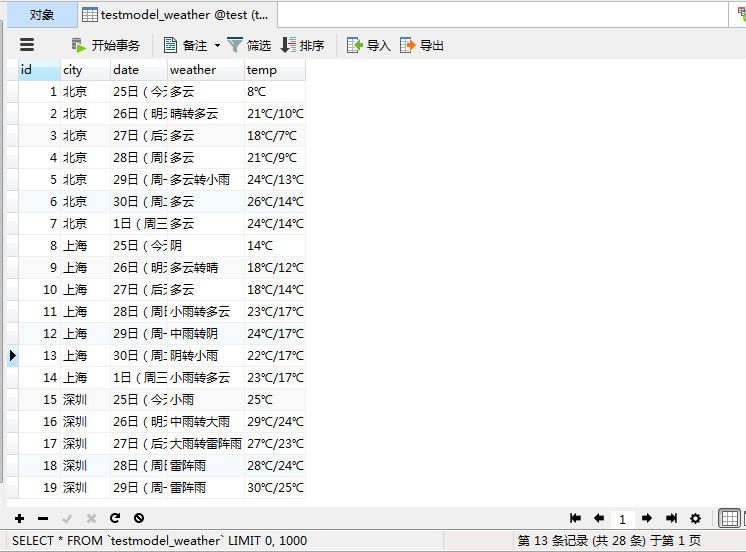
except Exception as err:

print(err)

cursor.close()

conn.commit()

conn.close()

**数据存储在SQL中： Django界面显示截图：**



**3.爬取淘宝书包数据**

代码：

import re

import requests

import pymysql

conn=pymysql.connect(host='localhost',user='root',passwd='1234',db='test',charset='utf8')

cursor=conn.cursor()

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])

cursor.execute("insert into testmodel\_shubao(title,price) values(%s,%s)",(title,price))

except:

print("")

cursor.close()

conn.commit()

conn.close()

def printGoodsList(ilt):

tplt="{:4}\t{:8}\t{:16}"

print(tplt.format("序号","价格","商品名称"))

count=0

for g in ilt:

count=count+1

print(tplt.format(count,g[0],g[1]))

def main():

goods="书包"

depth=2

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)

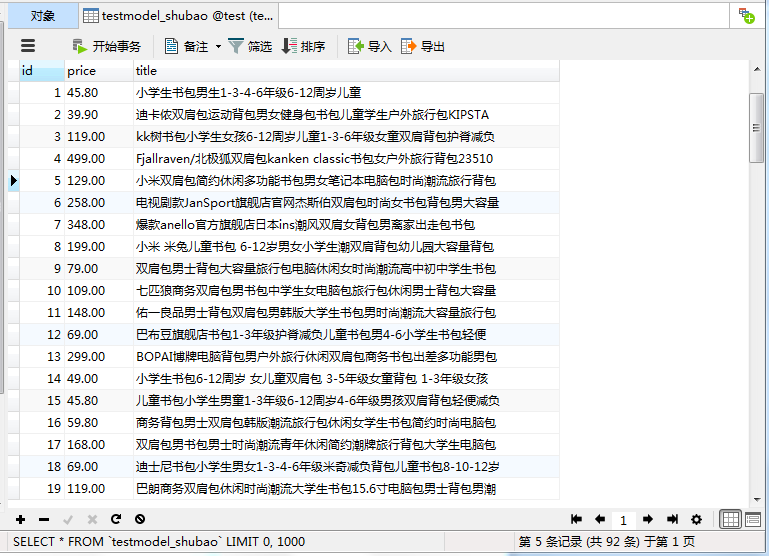
except:

continue

printGoodsList(infoList)

main()

**数据存储在SQL中： Django界面显示截图：**



**4.爬取京东手机数据**

代码：

from selenium import webdriver

from selenium.webdriver.chrome.options import Options

import urllib.request

import threading

import pymysql

import os

import datetime

from selenium.webdriver.common.keys import Keys

import time

class MySpider:

headers= {

"User-Agent": "Mozilla/5.0 (Windows; U; Windows NT 6.0 x64; en-US; rv:1.9pre)Gecko/2008072421 Minefield/3.0.2pre"

}

imagePath = "download"

def startUp(self,url,key):

chrome\_options=Options()

chrome\_options.add\_argument('--headless')

chrome\_options.add\_argument('--disable-gpu')

self.driver = webdriver.Chrome(chrome\_options=chrome\_options)

self.threads=[]

self.No=0

self.imgNo=0

try:

self.con=pymysql.connect(host='localhost',user='root',passwd=1234,db=test,charset="utf8")

self.cursor=self.con.cursor()

try:

self.cursor.execute("drop table testmodel\_phone ")

except:

pass

try:

sql="create table testmodel\_phone(mNo varchar(32) primary key,mMark varchar(256),mPrice varchar(32),mNote varchar(1024),mFile varchar(256))"

self.cursor.execute(sql)

except:

pass

except Exception as err:

print(err)

try:

if not os.path.exists(MySpider.imagePath):

os.mkdir(MySpider.imagePath)

images=os.listdir(MySpider.imagePath)

for img in images:

s=os.path.join(MySpider.imagePath,img)

os.remove(s)

except Exception as err:

print(err)

self.driver.get(url)

keyInput=self.driver.find\_element\_by\_id("key")

keyInput.send\_keys(key)

keyInput.send\_keys(Keys.ENTER)

def closeUp(self):

try:

self.con.commit()

self.con.close()

self.driver.close()

except Exception as err:

print(err)

def showDB(self):

try:

con=pymysql.connect(host='localhost',user='root',passwd='1597',db='pachong',charset="utf8")

cursor=con.cursor()

print("%-8s %-16s %-8s %-16s %s" % ("No","Mark","Price","Image","Note"))

cursor.execute("select mNo,mMark,mPrice,mFile,mNote from phones order by mNo")

rows=cursor.fetchall()

for row in rows:

print("%-8s %-16s %-8s %-16s %s" % (row[0],row[1],row[2],row[3],row[4]))

con.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(2)

print(self.driver.current\_url)

lis=self.driver.find\_elements\_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:

src2=""

try:

price=li.find\_element\_by\_xpath(".//div[@class='p-price']//i").text

except:

price="0"

try:

note=li.find\_element\_by\_xpath(".//div[@class='p-name p-name-type-2']//em").text

mark=note.split(" ")[0]

mark=mark.replace("爱心东东\n","")

mark=mark.replace(",","")

note=note.replace("爱心东东\n","")

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.cursor.execute("insert into jd\_test(mNo,mMark,mPrice,mNote,mFile) values(%s,%s,%s,%s,%s)",(no,mark,price,note,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("开始...")

self.startUp(url,key)

self.processSpider()

self.closeUp()

for t in self.threads:

t.join()

print("爬取完成....")

endtime=datetime.datetime.now()

elapsed=(endtime-starttime).seconds

print("共用",elapsed,"秒时间")

url="http://www.jd.com"

spider=MySpider()

while True:

print("1.爬取")

print("2.显示")

print("3.退出")

s=input("请选择(1,2,3):")

if s=="1":

spider.executeSpider(url,"手机")

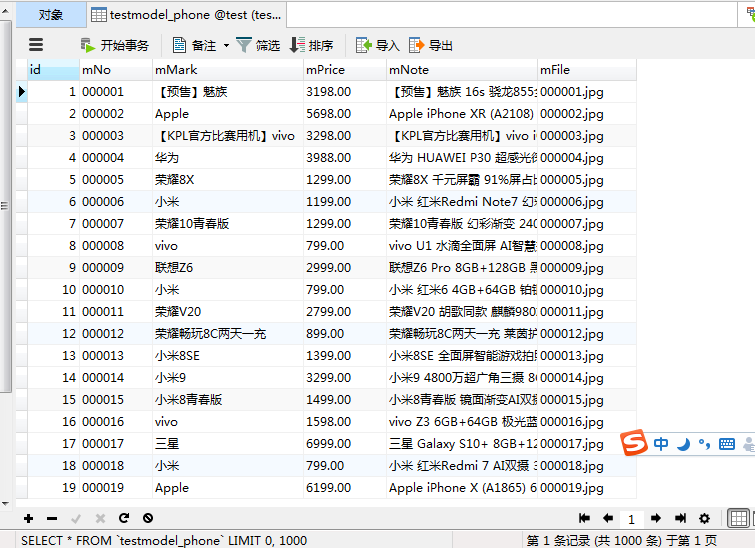
elif s=="2":

spider.showDB()

elif s=="3":

break

**数据存储在SQL中： Django界面显示截图：**





**（二）爬虫期末作业：**

**1. 通过selenium实现 12306自动登陆**

代码：

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.common.action\_chains import ActionChains

import requests

import base64

import re

import time

class Demo():

def \_\_init\_\_(self):

self.coordinate=[[-105,-20],[-35,-20],[40,-20],[110,-20],[-105,50],[-35,50],[40,50],[110,50]]

def login(self):

login\_url="https://kyfw.12306.cn/otn/resources/login.html"

driver = webdriver.Chrome()

driver.set\_window\_size(1200, 900)

driver.get(login\_url)

account=driver.find\_element\_by\_class\_name("login-hd-account")

account.click()

userName=driver.find\_element\_by\_id("J-userName")

userName.send\_keys("531218020@qq.com")

password=driver.find\_element\_by\_id("J-password")

password.send\_keys("\*\*\*\*\*\*\*\*\*\*\*")

self.driver=driver

def getVerifyImage(self):

try:

img\_element =WebDriverWait(self.driver, 100).until(

EC.presence\_of\_element\_located((By.ID, "J-loginImg"))

)

except Exception as e:

print("网络开小差,请稍后尝试")

base64\_str=img\_element.get\_attribute("src").split(",")[-1]

imgdata=base64.b64decode(base64\_str)

with open('verify.jpg','wb') as file:

file.write(imgdata)

self.img\_element=img\_element

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 e:

print(e)

def submit(self):

self.driver.find\_element\_by\_id("J-login").send\_keys(Keys.ENTER)

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(1000)

Demo()()

