

# 温州大学瓯江学院

WENZHOU UNIVERSITY OUJIANG COLLEGE



## 爬虫与数据分析期中考试

专    业    计算机科学与技术

---

课    程    爬虫与数据分析

---

# 爬虫与数据分析期中考试

## 目录

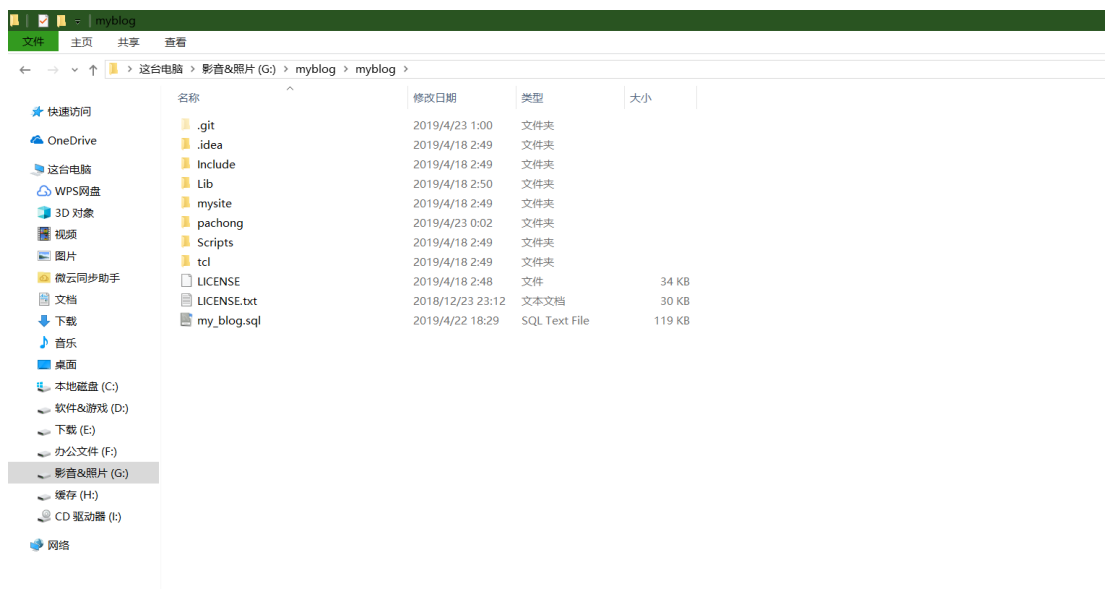
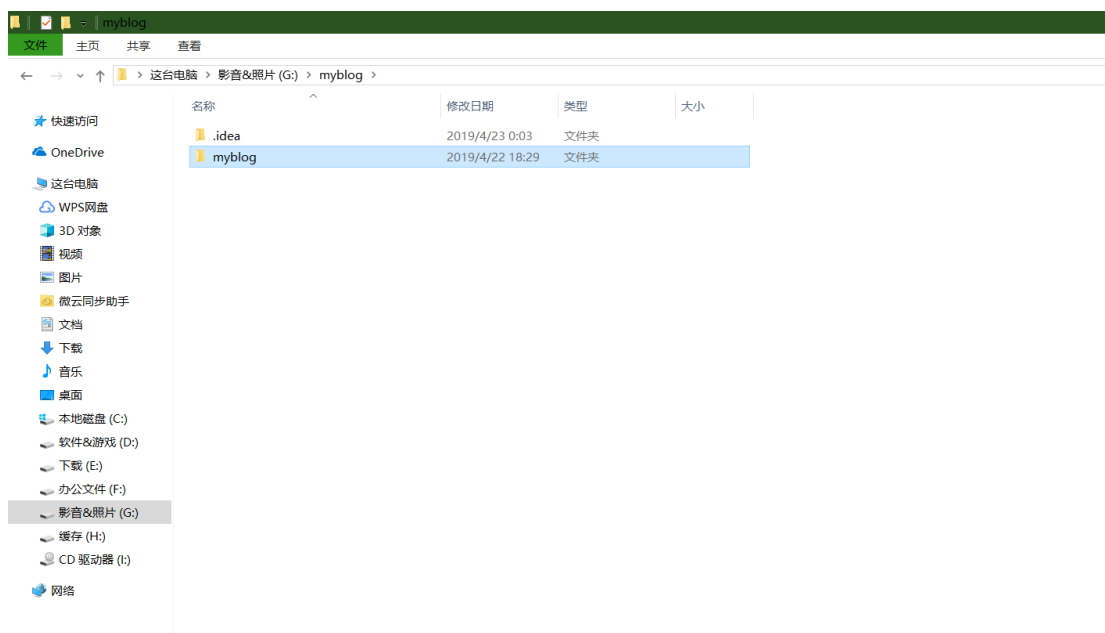
### 目录

WENZHOU UNIVERSITY OUJIANG COLLEGE .....	1
环境: .....	2
文档结构: .....	3
DY.py.....	5
JDphone.py.....	8
运行.....	13
博客首页: .....	16
登录: .....	18
后台: .....	19
爬虫显示: .....	22

## 环境:

Python3.7+pycharm2018.3+mysql

# 文档结构：

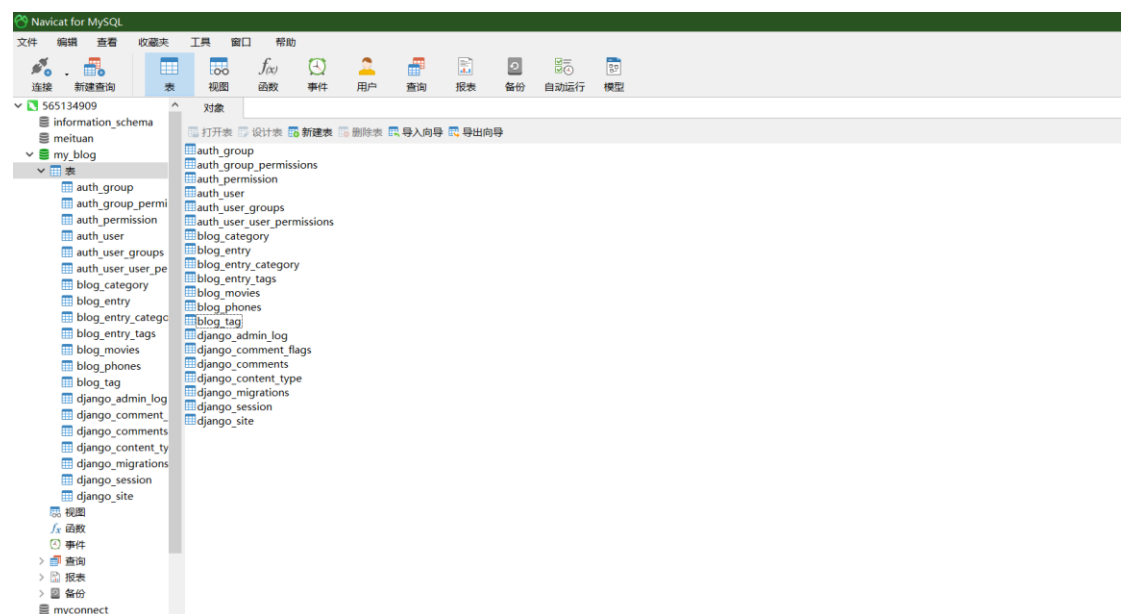


my\_blog.sql

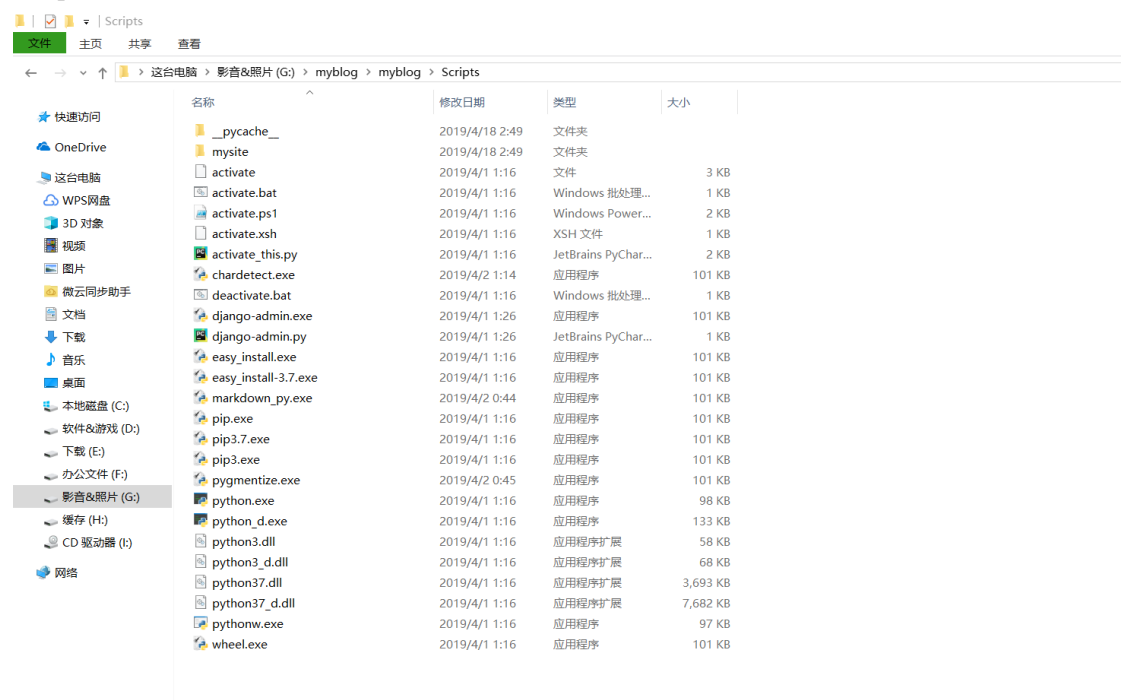
为对应的数据库文件，也可通过运行爬虫进行创建或通过

python manage.py migrate 重建表结构

数据库结构如下



## Scripts 文件夹为对应所需的虚拟环境



## pachong 文件夹下为对应的两个爬虫文件

DY.py

JDphone.py

# DY.py

```
# -*- coding:utf-8 -*-
import requests
import re
import mysql.connector

def changepage(url, total_page):
    page_group = ['https://www.dygod.net/html/gndy/jddy/index.html']
    for i in range(2, total_page + 1):
        link = re.sub('jddy/index', 'jddy/index_' + str(i), url, re.S)
        page_group.append(link)
    return page_group

def pagelink(url):
    base_url = 'https://www.dygod.net/html/gndy/jddy/'
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/45.0.2454.101 Safari/537.36'}
    req = requests.get(url, headers=headers)
    req.encoding = 'gbk'
    pat = re.compile('<a href="/html/gndy/jddy/(.*?)" class="ulink" title=(.*?)/a>', re.S)
    reslist = re.findall(pat, req.text)

    finalurl = []
    for i in range(1, 25):
        xurl = reslist[i][0]
        finalurl.append(base_url + xurl)
    return finalurl

def getdownurl(url):
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/45.0.2454.101 Safari/537.36'}
    req = requests.get(url, headers=headers)
    req.encoding = 'gbk'

    pat = re.compile('<a href="ftp(.*?)">ftp', re.S)
    reslist = re.findall(pat, req.text)
```

```
pat2 = re.compile('<!--Content Start-->(.*?)<!--duguPlayList Start-->', re.S)
reslist2 = re.findall(pat2, req.text)
reslist3 = re.sub('<[p></p>]', '', reslist2[0])
fdetail = reslist3.split('©')

return (furl, fdetail)
```

```
def inserttable(con, cs, x, y):  
    try:  
        cs.execute(  
            "insert into blog_movies values ('%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s',  
'%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s')\" \  
            % (x, y[0], y[1], y[2], y[3], y[4], y[5], y[6], y[7], y[8], y[9], y[10], y[11], y[12], y[13],  
y[14], y[15],  
            y[16], y[17]))  
    except:  
        pass  
    finally:  
        con.commit()
```

```
if __name__ == "__main__":
    html = "https://www.dygod.net/html/gndy/jddy/index.html"
    print('你即将爬取的网站是： https://www.dygod.net/html/gndy/jddy/index.html')
    pages = input('请输入需要爬取的页数： ')
    createtable
    p1 = changepage(html, int(pages))

    conn = mysql.connector.connect(host='localhost', user='root', password='123456',
```

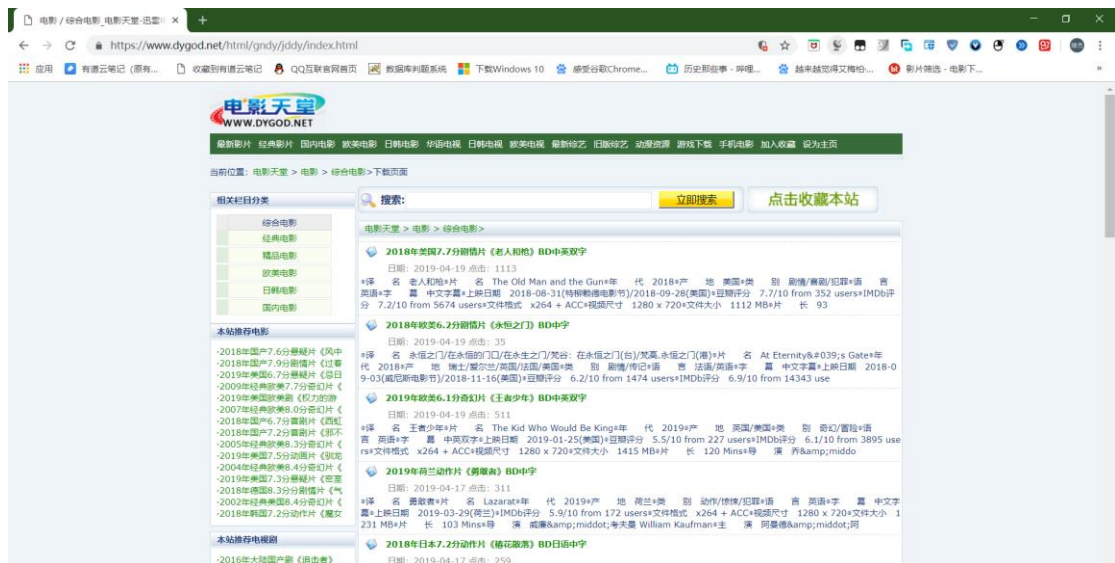
```

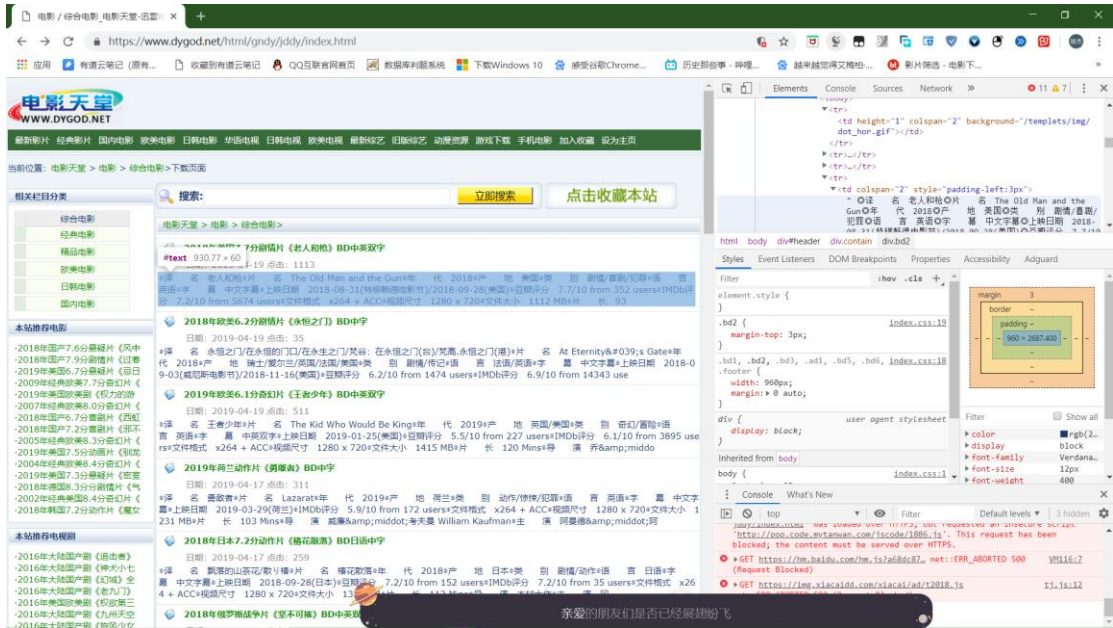
database='my_blog')
cursor = conn.cursor()
createtable(conn, cursor)
j = 0
for p1i in p1:
    j = j + 1
    print('正在爬取第%d 页,网址是 %s ...' % (j, p1i))
    p2 = pagelink(p1i)
    for p2i in p2:
        p3, p4 = getdownurl(p2i)
        if len(p3) == 0:
            pass
        else:
            inserttable(conn, cursor, p3, p4)
cursor.close()
conn.close()
print('所有页面地址爬取完毕!')

```

爬取的网站

<https://www.dygod.net/html/gndy/jddy/index.html>





用了正则表达式和 requests 对网页进行了爬取

## JDphone.py

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
import urllib.request
import threading
import mysql.connector
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(options=chrome_options)
self.threads = []
self.No = 0
self.imgNo = 0

try:
    self.con = mysql.connector.connect(host='localhost', user='root',
password='123456', database='my_blog')
    self.cursor = self.con.cursor()
    try:
        self.cursor.execute("drop table blog_phones")
    except:
        pass
    try:
        sql = "create table blog_phones (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 insertDB(self, mNo, mMark, mPrice, mNote, mFile):
    try:
        self.cursor.execute("insert into blog_phones values ('%s', '%s', '%s', '%s', '%s')"
\
                                % (mNo, mMark, mPrice, mNote, mFile))
    except Exception as err:
        print(err)

def showDB(self):
    try:
        con = mysql.connector.connect(host='localhost', user='root',
password='123456', database='my_blog')
        cursor = con.cursor()
        print("%-8s %-16s %-8s %-16s %s" % ("No", "Mark", "Price", "Image",
"Note"))
        cursor.execute("select mNo,mMark,mPrice,mFile,mNote from blog_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(1)
            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.insertDB(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("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, " seconds elapsed")

```

```

url = "http://www.jd.com"
spider = MySpider()
while True:
    print("1.爬取")
    print("2.显示")
    print("3.退出")
    s = input("请选择(1,2,3):")

```

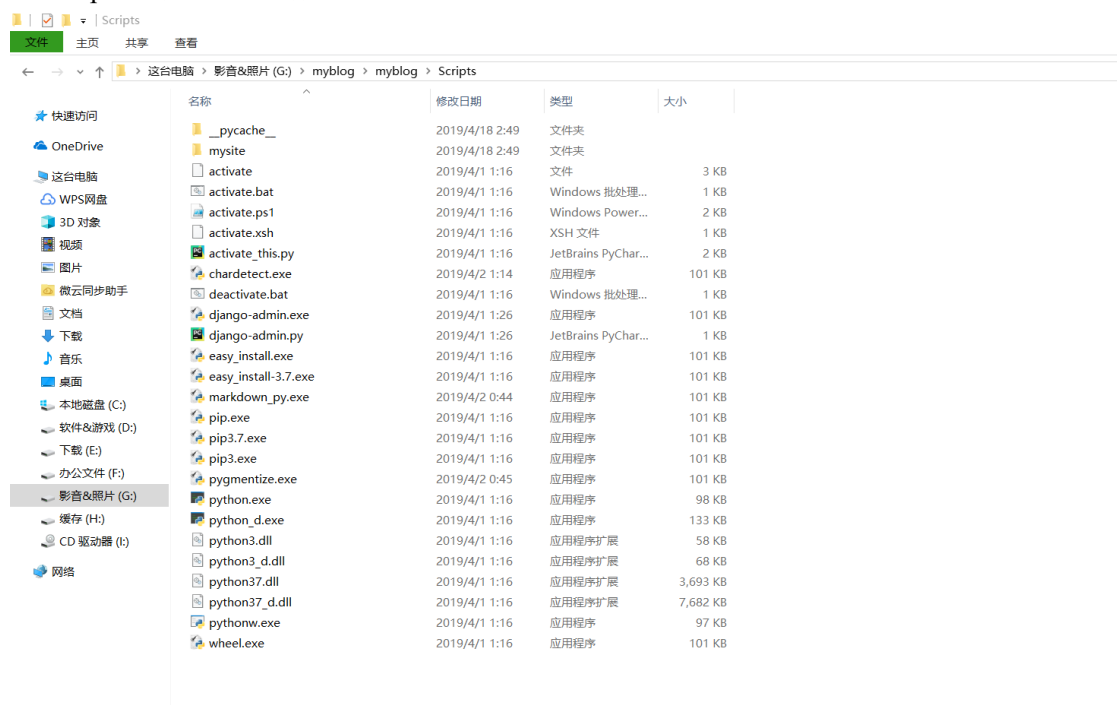
```
if s == "1":
    spider.executeSpider(url, "手机")
elif s == "2":
    spider.showDB()
elif s == "3":
    break
```

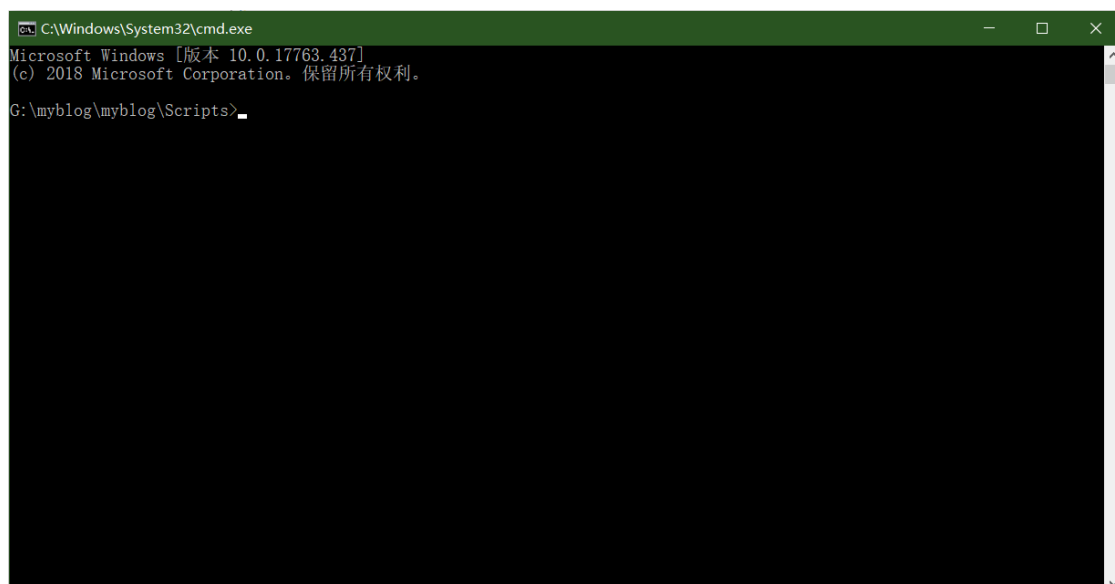
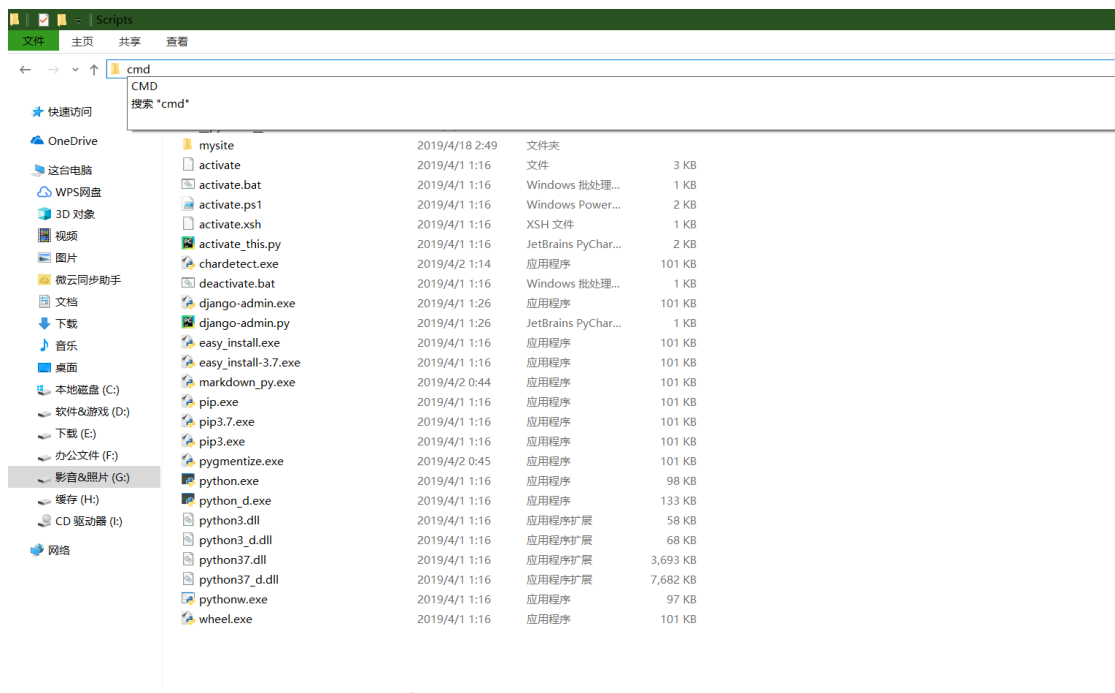
通过驱动使用无头浏览器对京东商城的手机进行爬取

download 通过两个路径对网页进行下载，提升了下载的成功几率

## 运行

在 Script 文件夹地址栏输入 cmd 运行





依次输入

activate

cd ../

cd mysite

创建超级管理员

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17763.437]
(c) 2018 Microsoft Corporation。保留所有权利。

G:\myblog\myblog\Scripts>activate

(myblog) G:\myblog\myblog\Scripts>cd ../

(myblog) G:\myblog\myblog>cd mysite

(myblog) G:\myblog\myblog\mysite>python manage.py createsuperuser

You have 1 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): blog.
Run 'python manage.py migrate' to apply them.
用户名 (leave blank to use 'empirestatebuilding'): 565134909
电子邮件地址: 565134909@qq.com
Password:
Password (again):
这个密码全部是数字的。
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.

(myblog) G:\myblog\myblog\mysite>
```

启动（使用前请确保已打开数据库）

python manage.py runserver

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17763.437]
(c) 2018 Microsoft Corporation。保留所有权利。

G:\myblog\myblog\Scripts>activate

(myblog) G:\myblog\myblog\Scripts>cd ../

(myblog) G:\myblog\myblog>cd mysite

(myblog) G:\myblog\myblog\mysite>python manage.py runserver
```

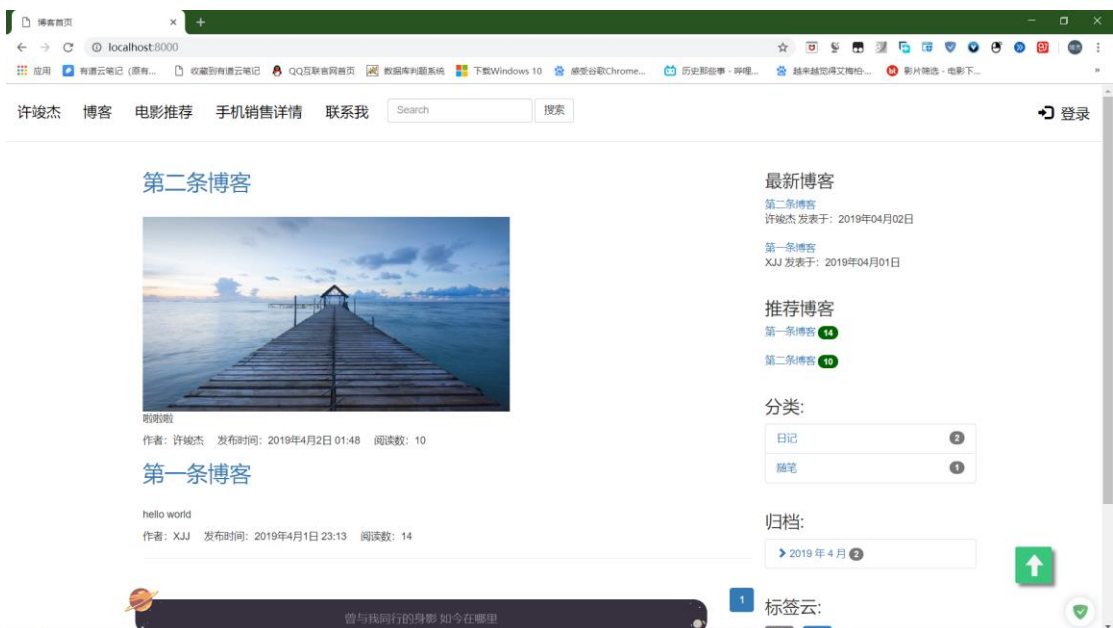
```
C:\Windows\System32\cmd.exe - python manage.py runserver
Microsoft Windows [版本 10.0.17763.437]
(c) 2018 Microsoft Corporation。保留所有权利。

G:\myblog\myblog\Scripts>activate
(myblog) G:\myblog\myblog\Scripts>cd ../
(myblog) G:\myblog\myblog>cd mysite
(myblog) G:\myblog\myblog\mysite>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 1 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): blog.
Run 'python manage.py migrate' to apply them.
April 23, 2019 - 01:56:24
Django version 2.1.7, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
_
```

## 博客首页：

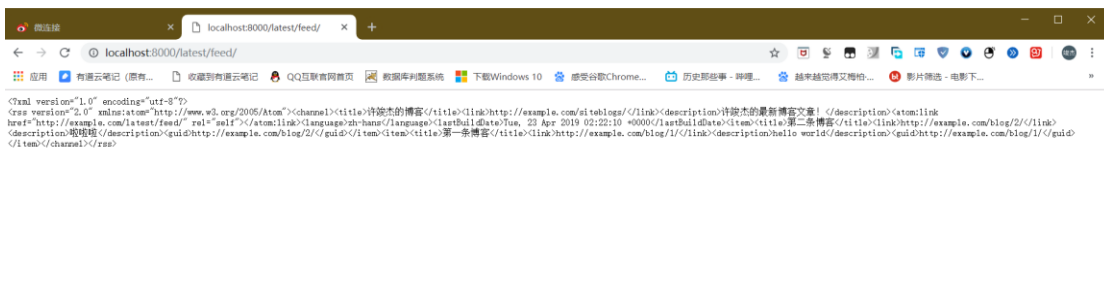


评论区：



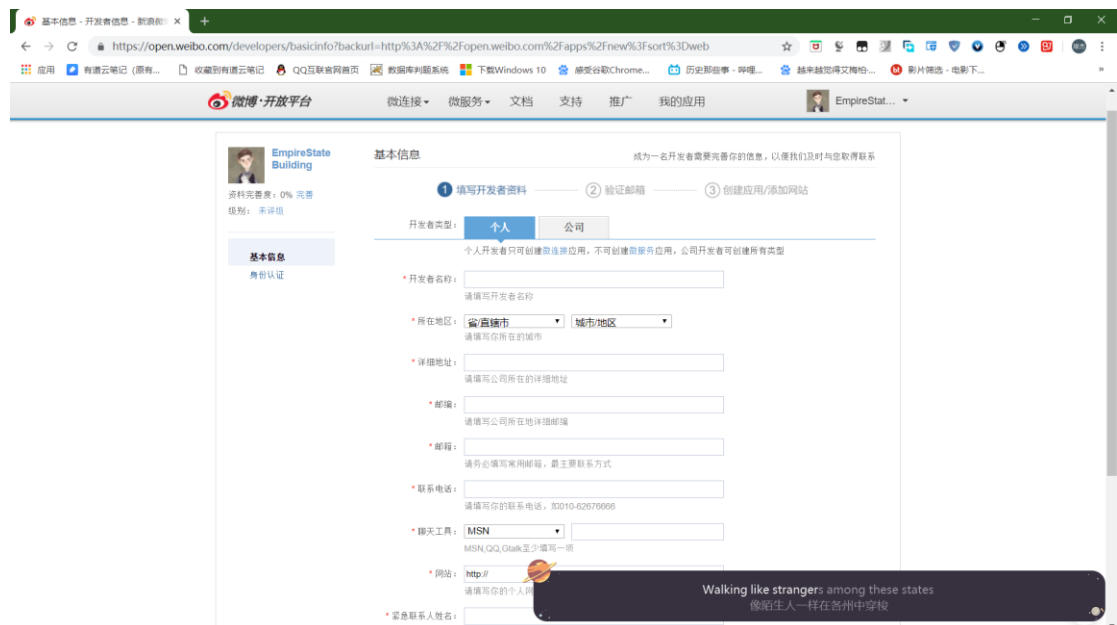


其他功能：阅读数统计，标签云，分类，回到顶部及博客的 xml 显示

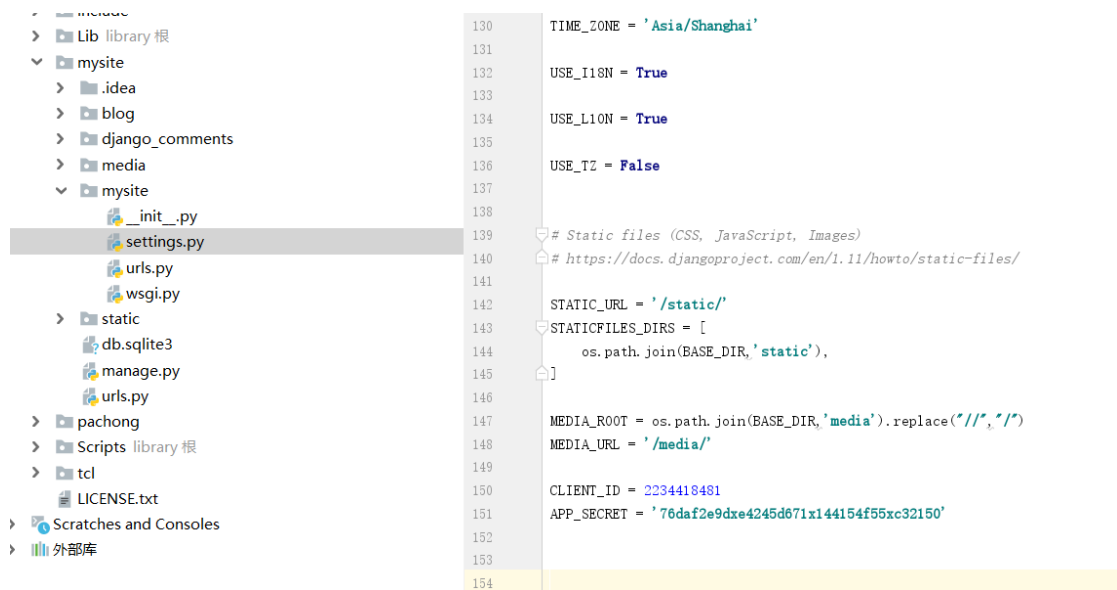


# 登录:

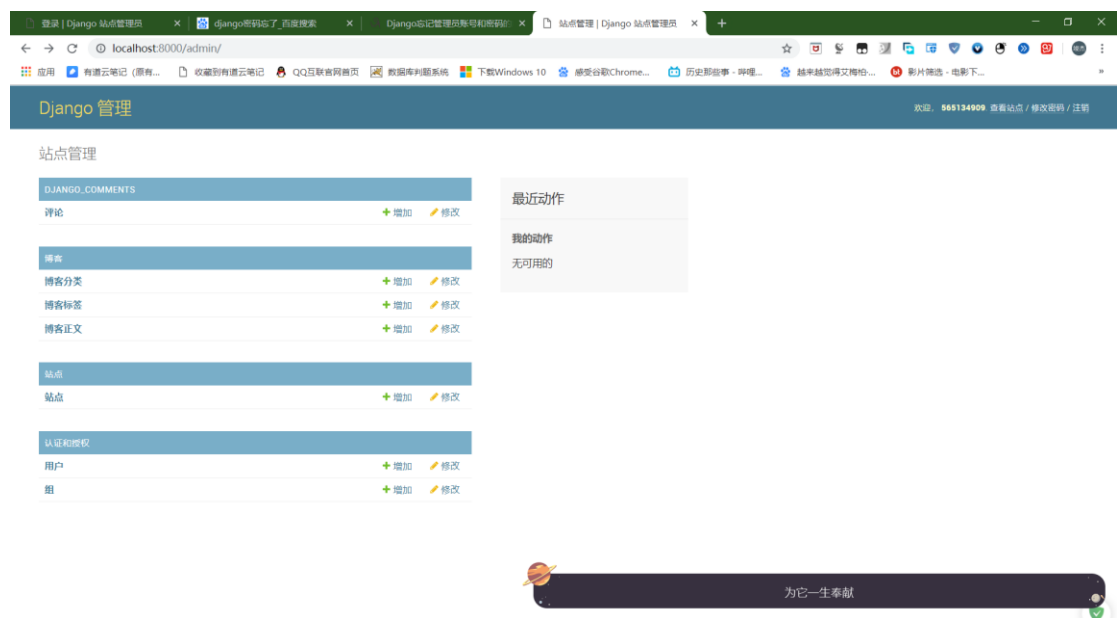
登录功能通过微博接入，但需要申请开发者，获得 `CLIENT_ID` 和 `APP_SECRET`  
但由于本项目并未部署到服务器，缺少网址，暂时无法申请  
如有需要可到 <https://open.weibo.com/>



用获得的 `CLIENT_ID` 和 `APP_SECRET`  
替换  
`mysite/mysite/setting.py` 里的 `CLIENT_ID` 和 `APP_SECRET` 即可





后台:



增加 博客正文

文章标题:


作者:   

博客配图:  未选择任何文件

正文:

摘要:

访问量:

博客分类:  



Django管理

首页 > Django.Comments > 增加评论


增加评论

内容类型:

对象ID:

现场:   

内容

用户:  

用户名:

用户图像:  未选择任何文件



用户的电子邮件地址:

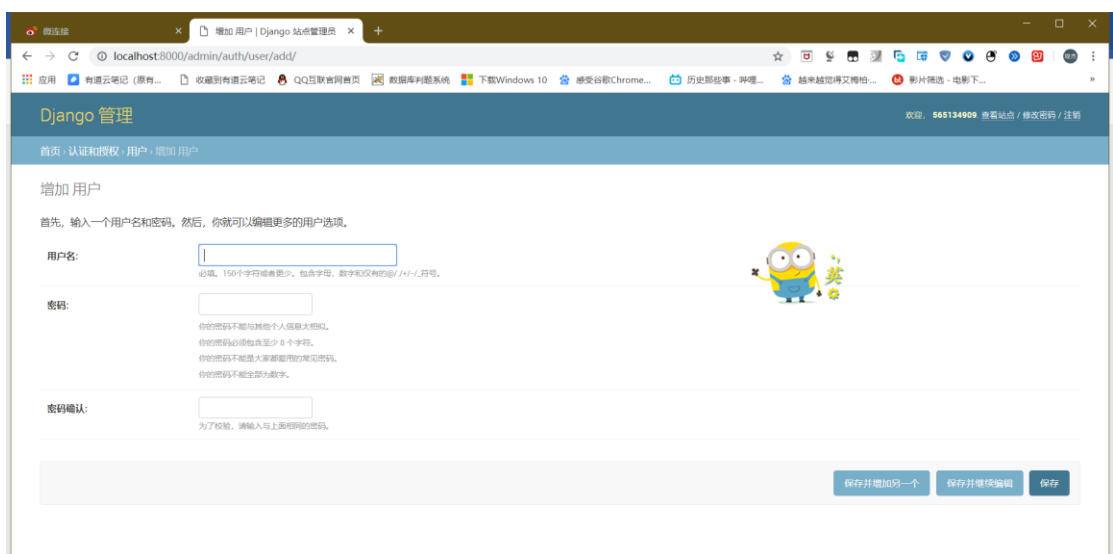
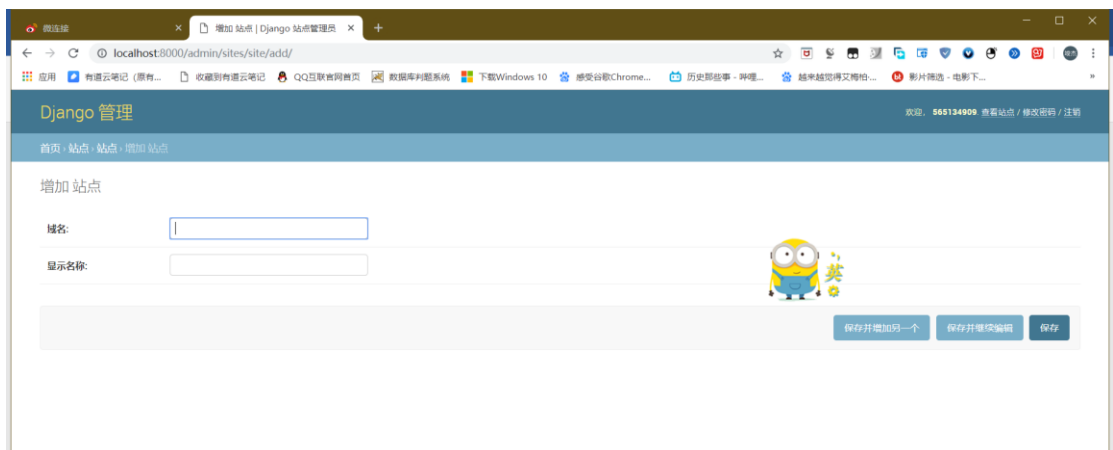
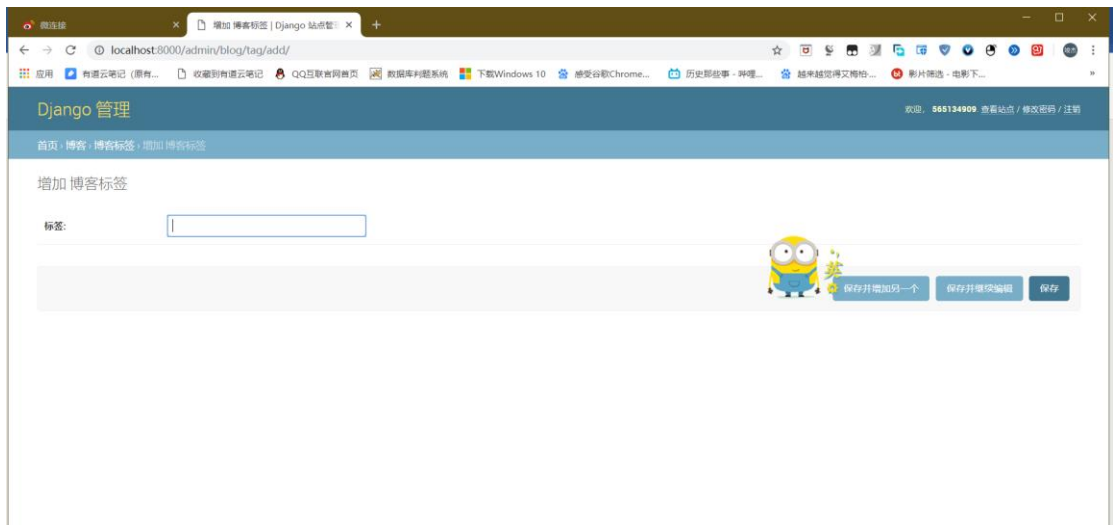
Django 管理

首页 > 博客 > 博客分类 > 增加 博客分类

增加 博客分类

分类:





# 爬虫显示:

