

学号: 16219111330
姓名: 王皓
班级: 16 计算机三班

项目基于 *django+bootstrap+selenium+bs4+mysql+request+python3.6*
Python 需要 pip 安装 *django,lxml,selenium,bs4,request,mysqlclient*,等运行库

| | |
|-----------------------|----|
| 项目内容结构: | 2 |
| 页面效果: | 3 |
| pc 端..... | 3 |
| 控制台功能..... | 5 |
| 指定爬取 | 5 |
| 获取所有 | 8 |
| 更新数据 | 9 |
| 代码说明 | 9 |
| 爬虫 get...py 文件: | 9 |
| View.py 搜索方法: | 9 |
| Control.py: | 10 |
| Index.html | 10 |

django 项目安装使用说明:

| django > python_crawler > python_crawler | | | |
|--|-----------------|-------------|----------|
| 名称 | 修改日期 | 类型 | 大小 |
| __pycache__ | 2019/4/21 13:08 | 文件夹 | |
| __init__.py | 2019/4/20 14:42 | Python File | 0 KB |
| chromedriver.exe | 2019/4/20 15:09 | 应用程序 | 8,348 KB |
| control.py | 2019/4/21 12:30 | Python File | 2 KB |
| getmovies.py | 2019/4/20 18:04 | Python File | 2 KB |
| getphones.py | 2019/4/21 0:22 | Python File | 3 KB |
| getweathers.py | 2019/4/21 13:08 | Python File | 2 KB |
| settings.py | 2019/4/21 13:01 | Python File | 4 KB |
| urls.py | 2019/4/21 13:02 | Python File | 2 KB |
| view.py | 2019/4/21 2:30 | Python File | 2 KB |
| wsgi.py | 2019/4/20 14:42 | Python File | 1 KB |

编辑 settings.py, 修改数据库配置

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'python_crawler', #数据库名
        'USER': 'sa', #用户名
        'PASSWORD': '1234', #密码
        'HOST': 'localhost', #地址
        'PORT': '3306', #端口
    }
}
```

然后用 cmd, cd 到项目根路径下输入

python manage.py migrate 运行 python

manage.py makemigrations crawler python

manage.py migrate crawler

创建表结构后

输入 python manage.py runserver +本机 ip+ --insecure 启动 django 项目

项目内容结构:

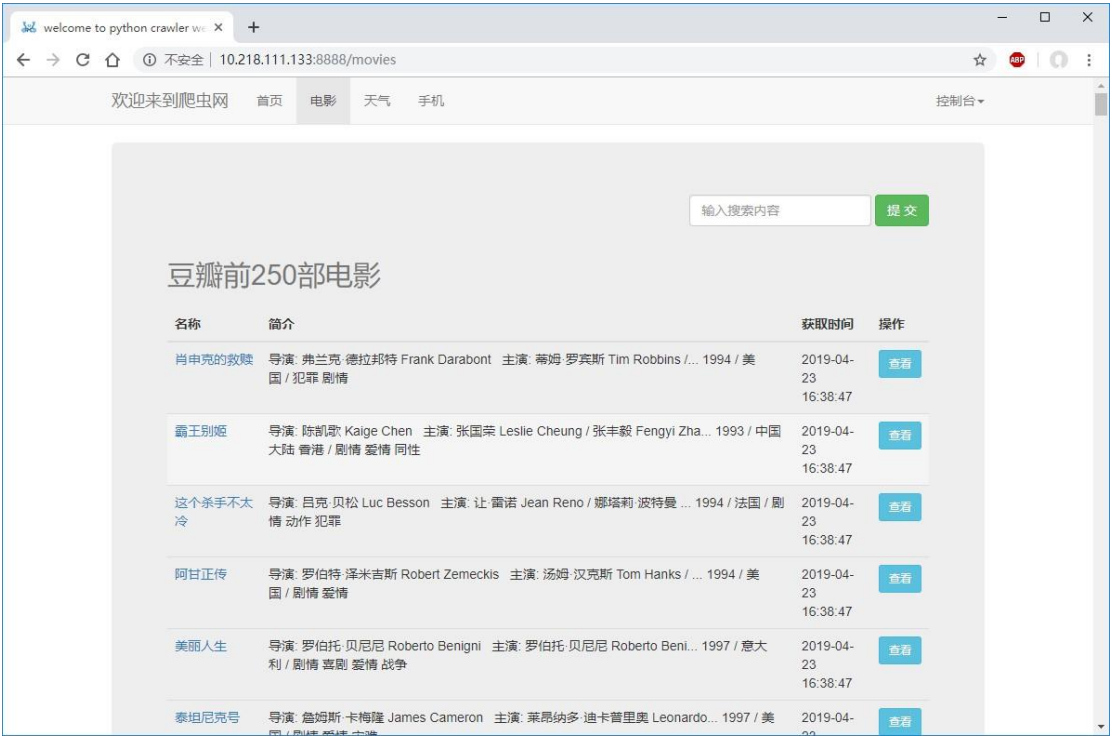
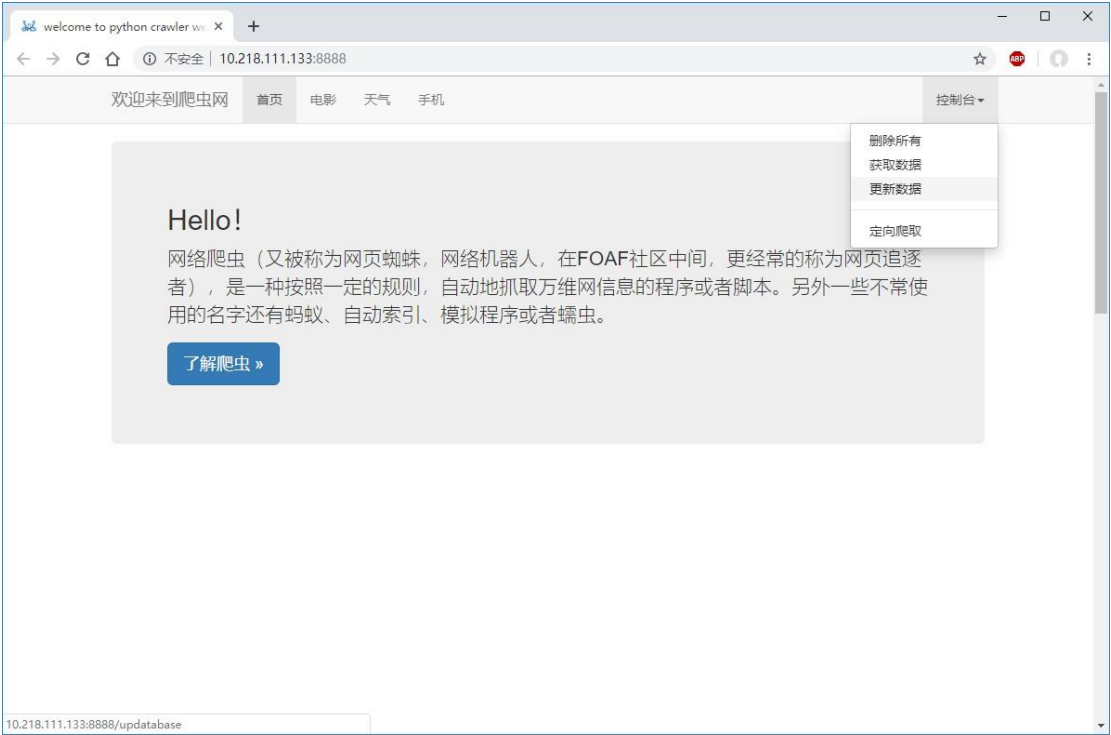
| | | | |
|----------------|-----------------|-------------|------|
| crawler | 2019/4/20 15:02 | 文件夹 | |
| python_crawler | 2019/4/20 17:03 | 文件夹 | |
| static | 2019/4/20 22:46 | 文件夹 | |
| templates | 2019/4/21 12:36 | 文件夹 | |
| manage.py | 2019/4/20 14:42 | Python File | 1 KB |

| | | | |
|------------------|-----------------|-------------|----------|
| __pycache__ | 2019/4/23 15:46 | 文件夹 | |
| __init__.py | 2019/4/20 14:42 | Python File | 0 KB |
| chromedriver.exe | 2019/4/20 15:09 | 应用程序 | 8,348 KB |
| control.py | 2019/4/21 12:30 | Python File | 2 KB |
| getmovies.py | 2019/4/20 18:04 | Python File | 2 KB |
| getphones.py | 2019/4/21 0:22 | Python File | 3 KB |
| getweathers.py | 2019/4/21 13:08 | Python File | 2 KB |
| settings.py | 2019/4/23 15:46 | Python File | 4 KB |
| urls.py | 2019/4/21 13:02 | Python File | 2 KB |
| view.py | 2019/4/21 2:30 | Python File | 2 KB |
| wsgi.py | 2019/4/20 14:42 | Python File | 1 KB |

三个 get 的 py 文件是爬虫文件, chromedriver.exe 是谷歌浏览器驱动, urls.py 绑定 url 与后台函数, view.py 处理前台页面内容, control.py 负责数据库爬虫操作

页面效果：

pc 端



提交

各地天气

| 所在位置 | 日期 | 风级 | 最低温度 | 最高温度 | 天气 |
|--------------|----------|----------|------|------|------|
| 黑龙江>哈尔滨> 城区 | 21日 (今天) | <3级 | 3℃ | 16 | 晴 |
| 吉林>长春> 城区 | 21日 (今天) | 3-4级 | 5℃ | 16 | 晴 |
| 辽宁>沈阳> 城区 | 21日 (今天) | 3-4级 | 4℃ | 19 | 晴 |
| 内蒙古>呼和浩特> 城区 | 21日 (今天) | 3-4级 | 11℃ | 23 | 晴 |
| 山西>太原> 城区 | 21日 (今天) | <3级 | 7℃ | 24 | 晴 |
| 陕西>西安> 城区 | 21日 (今天) | <3级 | 15℃ | 21 | 阴转多云 |
| 山东>济南> 城区 | 21日 (今天) | 3-4级转<3级 | 12℃ | 19 | 阴转多云 |

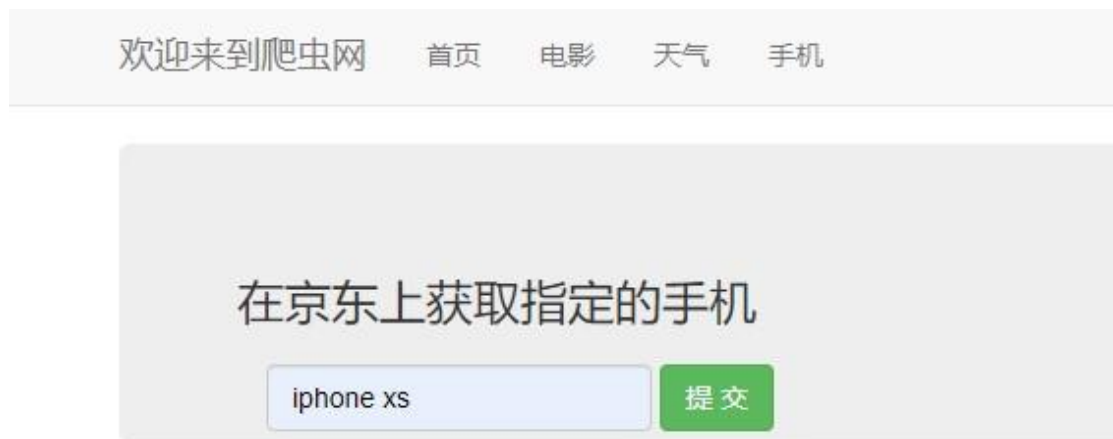
提交

京东手机

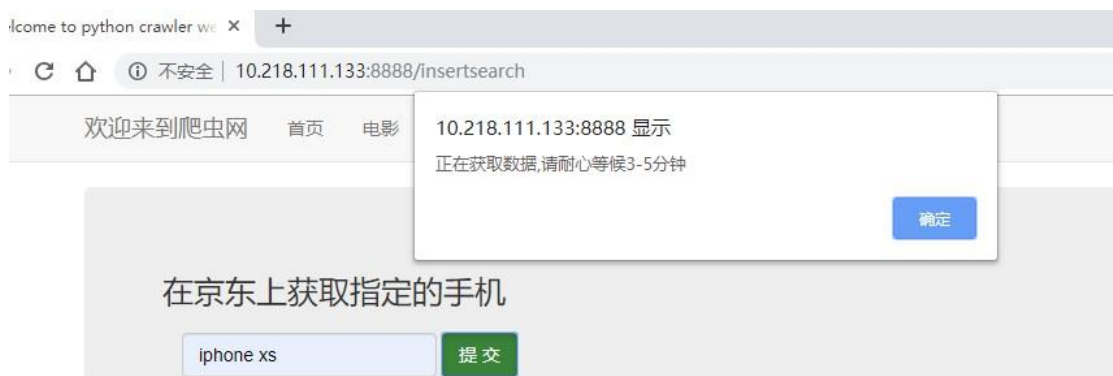
| 名称 | 价格 | 获取时间 | 操作 |
|---|-----------|---------------------|---------------|
| 华为 HUAWEI P30 超感光徕卡三摄麒麟980AI智能芯片全面屏屏内指纹版手机8GB+64GB亮黑色全网通4G手机双 | ¥ 3988.00 | 2019-04-23 15:24:16 | <div>查看</div> |
| 华为 HUAWEI P30 Pro 超感光徕卡四摄10倍混合变焦麒麟980芯片屏内指纹 8GB+128GB亮黑色全网通版双4G手机 | ¥ 4288.00 | 2019-04-23 15:24:16 | <div>查看</div> |
| 华为P30 手机【免息送6件豪礼华为原厂直供现货速发】亮黑色 全网通 (8GB+128GB) 屏内指纹 | ¥ 6788.00 | 2019-04-23 15:24:16 | <div>查看</div> |

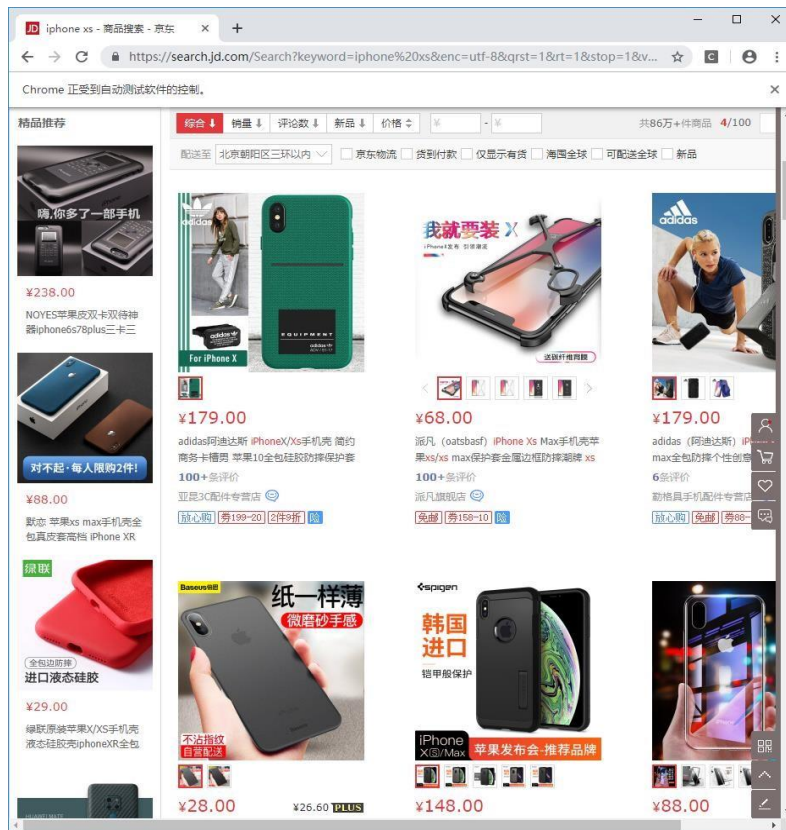
控制台功能

指定爬取



...之后可以加入更多搜索获取功能





欢迎来到爬虫网

首页

电影

天气

手机

iphone xs获取成功!

前往首页 »

iphone xs

提交

京东手机

| 名称 | 价格 | 获取时间 | 操作 |
|--|-----------|---------------------|----|
| Apple iPhone XS Max (A2104) 256GB 深空灰色 移动联通电信4G手机 双卡双待 | ¥ 9699.00 | 2019-04-23 16:32:28 | 查看 |
| Apple iPhone XS (A2100) 64GB 金色 移动联通电信4G手机 | ¥ 7599.00 | 2019-04-23 16:32:28 | 查看 |
| Apple 苹果 iPhone Xs 手机 金色 全网通 64GB | ¥ 7158.00 | 2019-04-23 16:32:28 | 查看 |

删除数据库

welcome to python crawler we

10.218.111.133:8888/phonesearch

欢迎来到爬虫网

首页

电影

天气

手机

控制台

删除所有

获取数据

更新数据

定向爬取

iphone xs

京东手机

| 名称 | 价格 | 获取时间 | 操作 |
|--|-----------|---------------------|----|
| Apple iPhone XS Max (A2104) 256GB 深空灰色 移动联通电信4G手机 双卡双待 | ¥ 9699.00 | 2019-04-23 16:32:28 | 查看 |
| Apple iPhone XS (A2100) 64GB 金色 移动联通电信4G手机 | ¥ 7599.00 | 2019-04-23 16:32:28 | 查看 |
| Apple 苹果 iPhone Xs 手机 金色 全网通 64GB | ¥ 7158.00 | 2019-04-23 16:32:28 | 查看 |
| Apple 苹果 iPhone Xs Max 手机 金色 全网通 256GB | ¥ 8788.00 | 2019-04-23 16:32:28 | 查看 |
| Apple 苹果 iPhone Xs 手机 全网通 金色 256GB | ¥ 8328.00 | 2019-04-23 16:32:28 | 查看 |
| Apple 苹果 iPhone Xs Max 手机 全网通 金色 256GB | ¥ 8758.00 | 2019-04-23 16:32:28 | 查看 |

10.218.111.133:8888/deleteall



获取所有

就是把三个爬虫代码执行一边并存储到数据库,,，自动化，一键即可

```
Quit the server with CTRL-BREAK.
电影爬取成功!

DevTools listening on ws://127.0.0.1:53451/devtools/browser/244e0f2f-e742-4373-8a26-0d63188bbc8e
<selenium.webdriver.remote.webelement.WebElement (session="ff88cc92b7e8ec07bc8f7f7ccf2e703b", element="0.14642428618281733-1")>
<selenium.webdriver.remote.webelement.WebElement (session="ff88cc92b7e8ec07bc8f7f7ccf2e703b", element="0.14642428618281733-2")>
<selenium.webdriver.remote.webelement.WebElement (session="ff88cc92b7e8ec07bc8f7f7ccf2e703b", element="0.14642428618281733-3")>
<selenium.webdriver.remote.webelement.WebElement (session="ff88cc92b7e8ec07bc8f7f7ccf2e703b", element="0.14642428618281733-4")>
<selenium.webdriver.remote.webelement.WebElement (session="ff88cc92b7e8ec07bc8f7f7ccf2e703b", element="0.14642428618281733-5")>
手机爬取成功!
```

豆瓣前 250 部电影, 京东 5 页手机商品, 天气是 101110101 到 101990101 编码的位置天气

```
http://www.weather.com.cn/weather/101'+str(i)+str(j)+'0101.shtml
```


更新数据

就是把数据库删除再获取，，emm，没啥操作

代码说明

爬虫 **get...py** 文件：

```
import MySQLdb from
crawler.models import Phones
def
get_phones(str):
    #
    conn=MySQLdb.connect(host="localhost",user="sa",passwd="1234",db="python_crawler",charset="utf8")    # cur=conn.cursor()
```

引入了 django 的模型，所以无需配置数据库连接，直接在 `setting.py` 修改即可，不过就导致无法本地运行，要直接运行要删除模型导入，并把 `conn` 和 `cur` 的注释取消

```
record=Phones(name=Name,url=Href,price=Price,time=now)
record.save()
        # cur.execute("insert into
crawler_phones(name,url,price,time)
values(%s,%s,%s,%s)",(Name,Href,Price,now))
```

删除 `record`，并把 `cur` 的注释取消

View.py 搜索方法：

```
def phonestearch(request):
    context = {}
    if request.POST:
        context['val'] = request.POST['val']

    phones=Phones.objects.filter(Q(name__icontains=context['val'])|Q(price__icontains=context['val'])|Q(time__icontains=context['val']))
    return render(request, 'phones.html', {'phones':
phones,'history':context['val']})
```

`__icontains` 类似于 `sql` 的 `like`，“注意是下划线”，下划线前是字段名
`Q` 对象| 等于 `or`

Control.py:

调用一下爬虫文件的方法

```
from . import
getmovies,getphones,getweathers,view import time
from crawler.models import Movies from
crawler.models import Weathers from
crawler.models import Phones from
django.shortcuts import render from django.http
import HttpResponse
def
deletelall(request):
    context = {}
try:
    Movies.objects.all().delete()
    Weathers.objects.all().delete()
Phones.objects.all().delete()
context['zhuangtai']='删除成功! ' except:
    context['zhuangtai']='删除失败! ' return
render(request, 'zhuangtai.html', context)
def
insertdata(request):
    context = {}
try:
    getmovies.get_movies()
time.sleep(2)
getphones.get_phones('小米9')
time.sleep(2)
getweathers.get_weather()
time.sleep(2)
context['zhuangtai']='获取成功! '
except:
    context['zhuangtai']='获取失败! ' return
render(request, 'zhuangtai.html', context)
```

Index.html

```
<li class="active"><a href="{% url 'index' %}">首页</a></li>
    <li><a href="{% url 'movies' %}">电影</a></li>
    <li><a href="{% url 'weathers' %}">天气</a></li>
```

手机 链接使用 urls.py 文件里的链接名，不写死链接方便 ip 端口更改

```
from django.conf.urls import *
from . import view, control

urlpatterns = [
    url(r'^$', view.index),
    url(r'^index$', view.index, name='index'),
    url(r'^movies$', view.movies, name='movies'),
    url(r'^weathers$', view.weathers, name='weathers'),
    url(r'^phones$', view.phones, name='phones'),
```