温州大學甌江學院
WENZHOU UNIVERSITY OUJIANG COLLEGE

《爬虫期中作业》

题　　目：	爬虫期中作业
分　　院：	数学与信息工程学院
班　　级：	16 计算机科学与技术三班
姓　　名：	葛佳俊
学　　号：	16219111322
完成日期：	2019 年 4 月 24 日

温州大学瓯江学院教务部

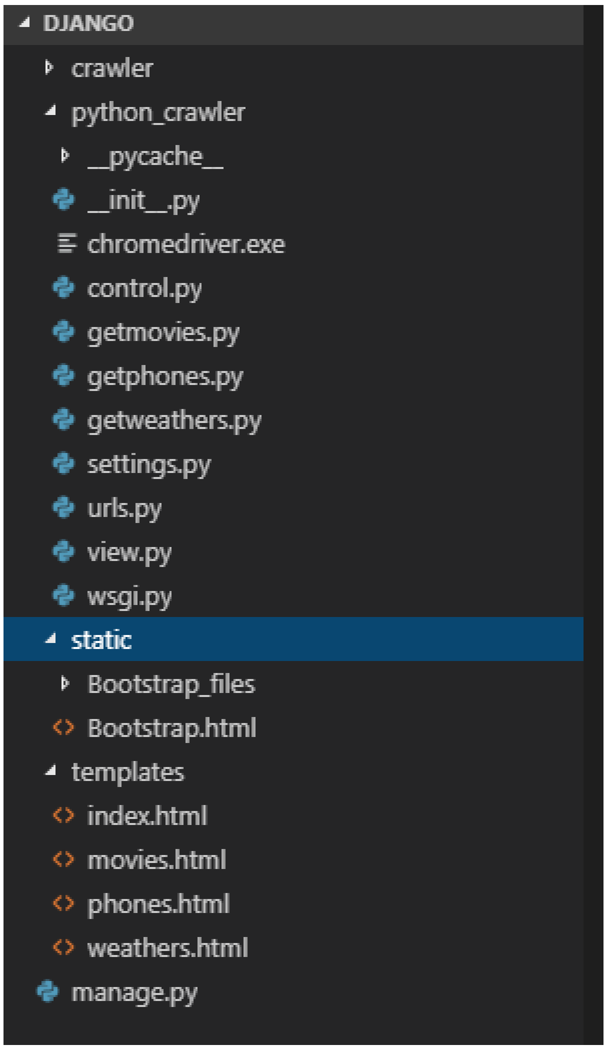
二〇一九年四月制

实验环境

环境：VS code 编辑器,Django,Python3.5,Mysql,bootstrap

Python 需要安装 django,lxml,selenium,bs4,requests,mysqlclient 等第三方库
如:pip install requests

项目结构



```
DJANGO
  crawler
  python_crawler
    __pycache__
    __init__.py
    chromedriver.exe
    control.py
    getmovies.py
    getphones.py
    getweathers.py
    settings.py
    urls.py
    view.py
    wsgi.py
  static
    Bootstrap_files
    Bootstrap.html
  templates
    index.html
    movies.html
    phones.html
    weathers.html
  manage.py
```

三个 getmovies.py,getphones.py,getweathers.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

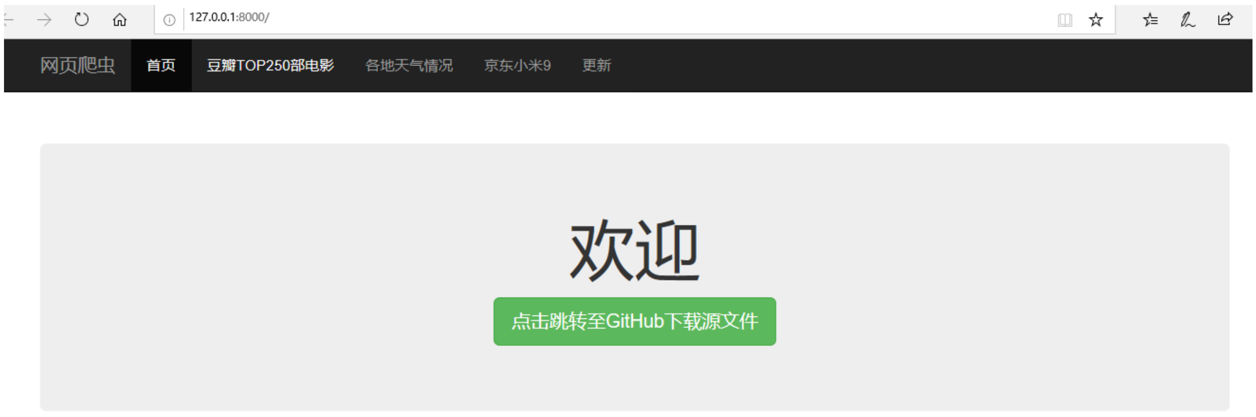
Dejango 配置

创建好 django 项目后打开 setting.py 文件,编辑数据库配置

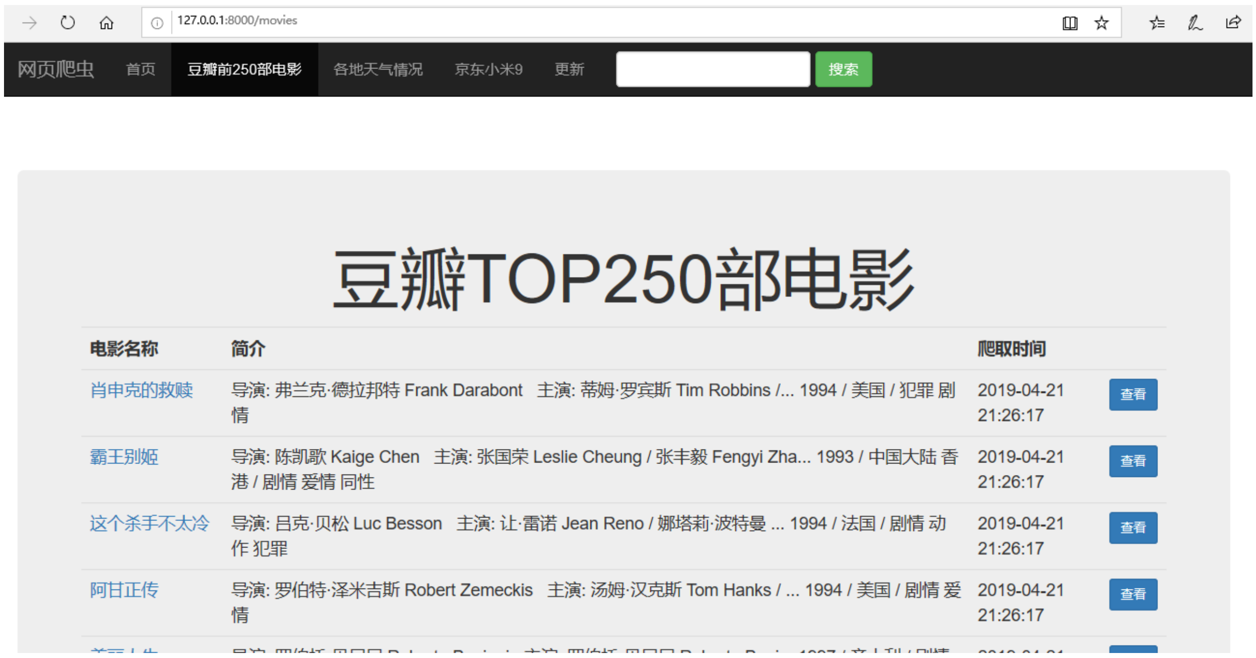
```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'test1',#数据库名
        'USER': 'root',#用户名
        'PASSWORD': 'gjj8897',#密码
        'HOST':'localhost',#地址
        'PORT':'3306',#端口
    }
}
```
使用 cmd 命令，cd 到项目根目录
运行命令 python manage.py migrate 创建相关数据表
输入 python manage.py runserver +本机 ip+ --insecure 即可启动 django 项目

页面效果展示

首页



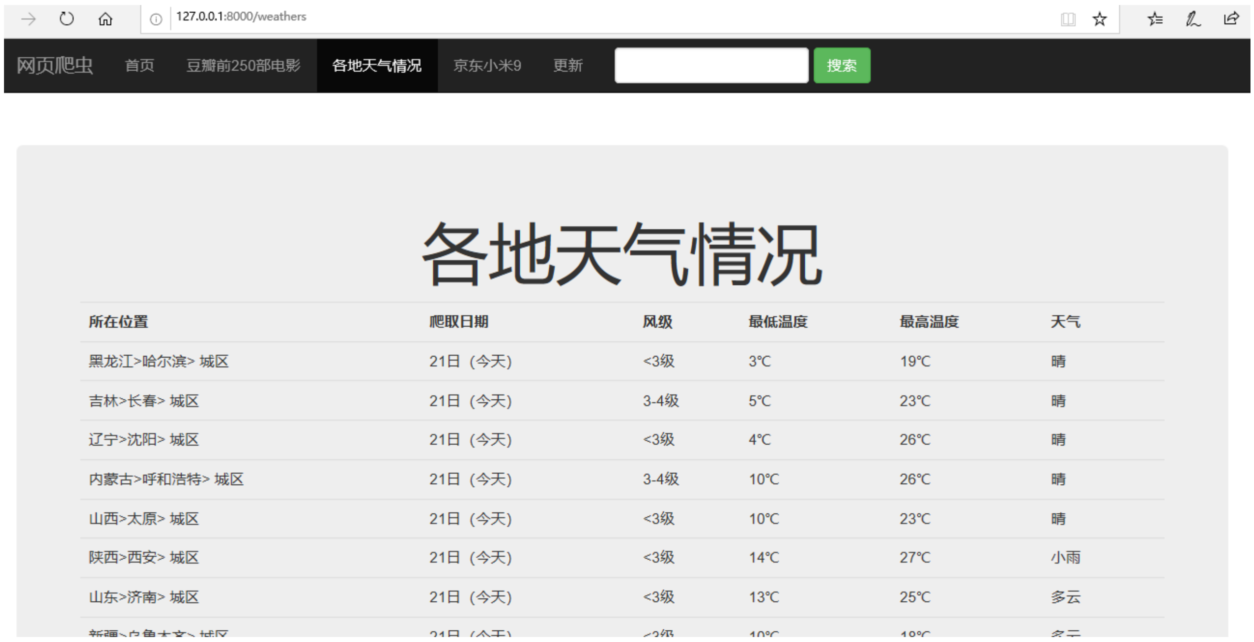
电影页



豆瓣TOP250部电影

电影名称	简介	爬取时间	
肖申克的救赎	导演: 弗兰克·德拉邦特 Frank Darabont 主演: 蒂姆·罗宾斯 Tim Robbins /... 1994 / 美国 / 犯罪 剧情	2019-04-21 21:26:17	查看
霸王别姬	导演: 陈凯歌 Kaige Chen 主演: 张国荣 Leslie Cheung / 张丰毅 Fengyi Zha... 1993 / 中国大陆 香港 / 剧情 爱情 同性	2019-04-21 21:26:17	查看
这个杀手不太冷	导演: 吕克·贝松 Luc Besson 主演: 让·雷诺 Jean Reno / 娜塔莉·波特曼 ... 1994 / 法国 / 剧情 动作 犯罪	2019-04-21 21:26:17	查看
阿甘正传	导演: 罗伯特·泽米吉斯 Robert Zemeckis 主演: 汤姆·汉克斯 Tom Hanks / ... 1994 / 美国 / 剧情 爱情	2019-04-21 21:26:17	查看

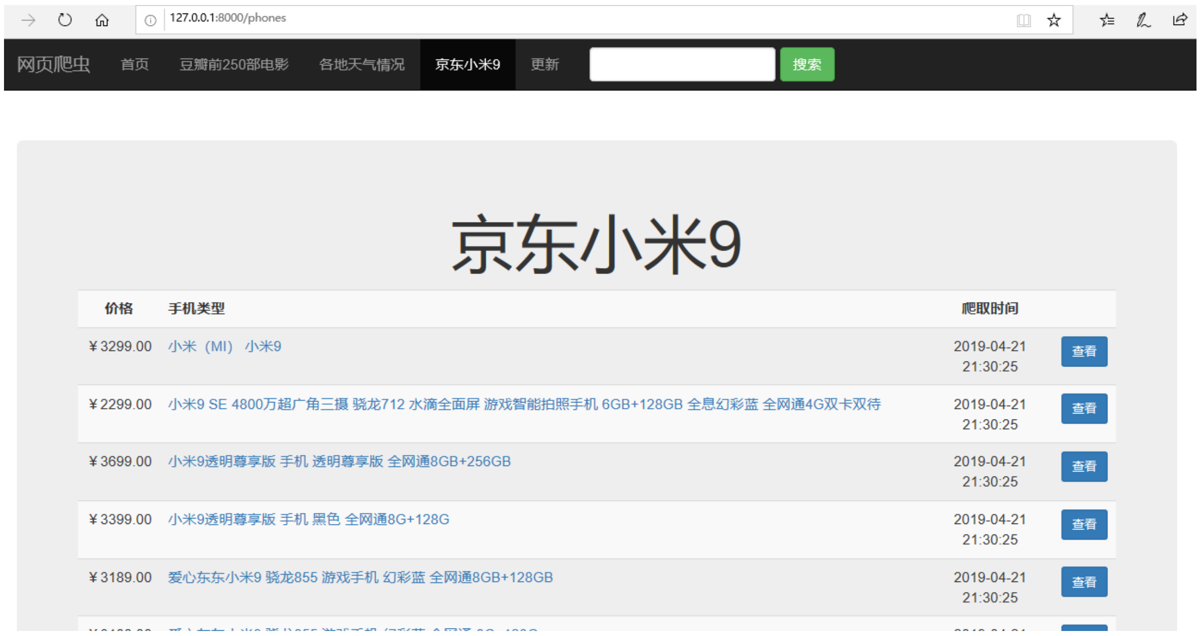
天气页


127.0.0.1:8000/weathers

各地天气情况

所在位置	爬取日期	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℃	19℃	晴
吉林>长春> 城区	21日（今天）	3-4级	5℃	23℃	晴
辽宁>沈阳> 城区	21日（今天）	<3级	4℃	26℃	晴
内蒙古>呼和浩特> 城区	21日（今天）	3-4级	10℃	26℃	晴
山西>太原> 城区	21日（今天）	<3级	10℃	23℃	晴
陕西>西安> 城区	21日（今天）	<3级	14℃	27℃	小雨
山东>济南> 城区	21日（今天）	<3级	13℃	25℃	多云

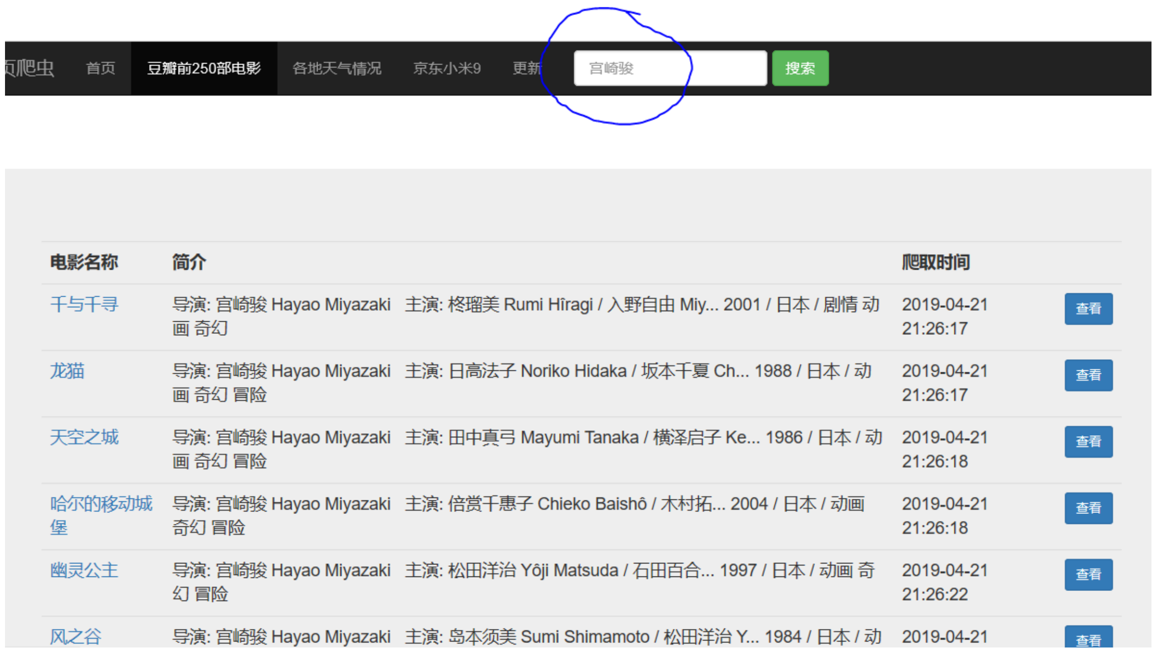
手机页


127.0.0.1:8000/phones

京东小米9

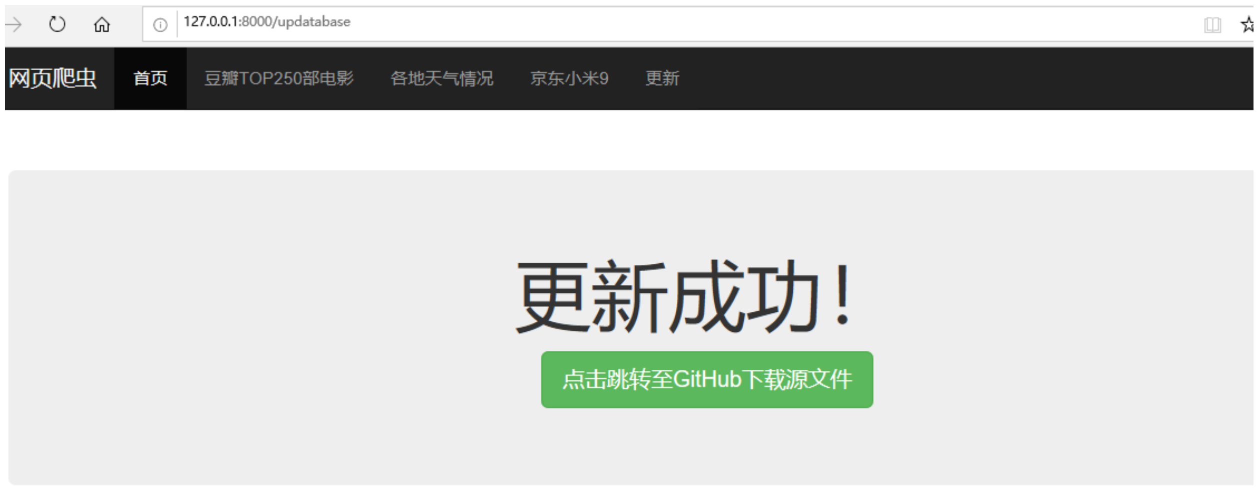
价格	手机类型	爬取时间	
¥3299.00	小米（MI） 小米9	2019-04-21 21:30:25	查看
¥2299.00	小米9 SE 4800万超广角三摄 骁龙712 水滴全面屏 游戏智能拍照手机 6GB+128GB 全息幻彩蓝 全网通4G双卡双待	2019-04-21 21:30:25	查看
¥3699.00	小米9透明尊享版 手机 透明尊享版 全网通8GB+256GB	2019-04-21 21:30:25	查看
¥3399.00	小米9透明尊享版 手机 黑色 全网通8G+128G	2019-04-21 21:30:25	查看
¥3189.00	爱心东东小米9 骁龙855 游戏手机 幻彩蓝 全网通8GB+128GB	2019-04-21 21:30:25	查看

根据关键字搜索当前页内容



电影名称	简介	爬取时间	
千与千寻	导演: 宫崎骏 Hayao Miyazaki 主演: 柊瑠美 Rumi Hîragi / 入野自由 Miy... 2001 / 日本 / 剧情 动画 奇幻	2019-04-21 21:26:17	查看
龙猫	导演: 宫崎骏 Hayao Miyazaki 主演: 日高法子 Noriko Hidaka / 坂本千夏 Ch... 1988 / 日本 / 动画 奇幻 冒险	2019-04-21 21:26:17	查看
天空之城	导演: 宫崎骏 Hayao Miyazaki 主演: 田中真弓 Mayumi Tanaka / 横泽启子 Ke... 1986 / 日本 / 动画 奇幻 冒险	2019-04-21 21:26:18	查看
哈尔的移动城堡	导演: 宫崎骏 Hayao Miyazaki 主演: 倍赏千惠子 Chieko Baishô / 木村拓... 2004 / 日本 / 动画 奇幻 冒险	2019-04-21 21:26:18	查看
幽灵公主	导演: 宫崎骏 Hayao Miyazaki 主演: 松田洋治 Yôji Matsuda / 石田百合... 1997 / 日本 / 动画 奇幻 冒险	2019-04-21 21:26:22	查看
风之谷	导演: 宫崎骏 Hayao Miyazaki 主演: 岛本须美 Sumi Shimamoto / 松田洋治 Y... 1984 / 日本 / 动	2019-04-21	查看

点击更新，删除数据库内容，重新爬取数据



Django 源代码

爬虫以爬取豆瓣 TOP250 电影为例

getmovies.py

```
import requests
from bs4 import BeautifulSoup
import MySQLdb
import time
from crawler.models import Movies
def get_movies():

#conn=MySQLdb.connect(host="localhost",user="root",passwd="gjj8897",db="python_crawler",charset="utf8")
```

```
    #cur=conn.cursor()
    headers={'user-agent':'Mozilla/5.0 (Windows NT 6.1;
Win64;x64)AppleWebKit/537.36 (KHTML,like Gecko) Chrome/52..02743.82
Safari/537.36','Host':'movie.douban.com'}
    for i in range(0,10):
        link='https://movie.douban.com/top250?start='+str(i*25)
        r=requests.get(link,headers=headers,timeout=10)
        soup=BeautifulSoup(r.text,"lxml")
        div_list=soup.find_all('div',class_='info')
        for each in div_list:
            url=each.div.a['href']
            title=each.div.a.span.text.strip()
            synopsis=each.contents[3].p.get_text().strip()
            now=time.strftime('%Y-%m-%d %H:%M:%S',time.localtime(time.time()))
            record=Movies(name=title,url=url,synopsis=synopsis,time=now)
            record.save()
        #     cur.execute("insert into crawler_movies(name,url,synopsis,time)
values(%s,%s,%s,%s)",(title,url,synopsis,now))
    print("电影爬取成功！")
#     cur.close()
#     conn.commit()
#     conn.close()
```

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，但也因此无法本地运行，若要直接 python 运行要删除模型导入 from crawler.models import Movies，并把 conn 和 cur 的注释取消，删除 record

view.py

```
from django.shortcuts import render
from django.http import HttpResponse
from crawler.models import Movies
from crawler.models import Weathers
from crawler.models import Phones
from django.db.models import Q

def index(request):
    context = {}
    context['hello']='欢迎'
    return render(request, 'index.html', {'hello':'欢迎'})

def movies(request):
    movies=Movies.objects.all()
    return render(request, 'movies.html',{'movies': movies,'hello':'豆瓣 TOP250 部电
影'})

def weathers(request):
    weathers=Weathers.objects.all()
    return render(request, 'weathers.html', {'weathers': weathers,'hello':'各地天气
情况'})

def phones(request):
    phones=Phones.objects.all()
    return render(request, 'phones.html', {'phones': phones,'hello':'京东小米 9'})

def searchmovies(request):
    context={}
    if request.POST:
        context['key']=request.POST['key']

movies=Movies.objects.filter(Q(name__icontains=context['key'])|Q(synopsis__iconta
ins=context['key'])|Q(time__icontains=context['key']))
    return render(request, 'movies.html', {'movies':
movies,'searchkey':context['key']})

def searchphones(request):
    context={}
    if request.POST:
        context['key']=request.POST['key']

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

def searchweathers(request):
    context={}
    if request.POST:
        context['key']=request.POST['key']

weathers=Weathers.objects.filter(Q(city__icontains=context['key'])|Q(dates__icont
ains=context['key'])|Q(winL__icontains=context['key'])

|Q(temperatureLow__icontains=context['key'])|Q(temperatureHigh__icontains=context
['key'])|Q(weather__icontains=context['key']))
    return render(request, 'weathers.html', {'weathers':
weathers,'searchkey':context['key']})
```

index.html
套用 bootstrap 现有模板

```
{% load staticfiles %}
<!DOCTYPE html>
<!-- saved from url=(0049)https://v3.bootcss.com/examples/starter-template/ -->
<html lang="zh-CN"><head><meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- 上述 3 个 meta 标签*必须*放在最前面，任何其他内容都*必须*跟随其后！ -->
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="icon" href="https://v3.bootcss.com/favicon.ico">

    <title>网页爬虫</title>

    <!-- Bootstrap core CSS -->
    <link href="{% static './Bootstrap_files/bootstrap.min.css' %}"
rel="stylesheet">

    <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
    <link href="{% static './Bootstrap_files/ie10-viewport-bug-workaround.css' %}"
rel="stylesheet">

    <!-- Custom styles for this template -->
    <link href="{% static './Bootstrap_files/starter-template.css' %}"
rel="stylesheet">

    <!-- Just for debugging purposes. Don't actually copy these 2 lines! -->
    <!--[if lt IE 9]><script
src="../../assets/js/ie8-responsive-file-warning.js"></script><![endif]-->
    <script src="{% static
'./Bootstrap_files/ie-emulation-modes-warning.js' %}"></script>

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries
-->
    <!--[if lt IE 9]>
      <script
src="https://cdn.bootcss.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script
src="https://cdn.bootcss.com/respond.js/1.4.2/respond.min.js"></script>
```

```
    <![endif]-->
  </head>

  <body>

    <nav class="navbar navbar-inverse navbar-fixed-top">
      <div class="container">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed"
data-toggle="collapse" data-target="#navbar" aria-expanded="false"
aria-controls="navbar">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
          <a class="navbar-brand" >网页爬虫</a>
        </div>
        <div id="navbar" class="collapse navbar-collapse" >
          <ul class="nav navbar-nav">
            {% block head %}
            <li class="active"><a href="http://127.0.0.1:8000/index">首页</a></li>
            <li><a href="http://127.0.0.1:8000/movies">豆瓣 TOP250 部电影</a></li>
            <li><a href="http://127.0.0.1:8000/weathers">各地天气情况</a></li>
            <li><a href="http://127.0.0.1:8000/phones">京东小米 9</a></li>
            <li><a href="http://127.0.0.1:8000/updatabase">更新</a></li>
            {% endblock %}
          </ul>
        </div><!--/.nav-collapse -->
      </div>
    </nav>

    <div class="container" style="margin-top:50px">
      <div class="jumbotron" style="text-align:center;" >
      {% block mainbody %}
          <h1>{{hello}}</h1>
          <p><a class="btn btn-lg btn-success" href="" role="button">点击跳转至 GitHub
下载源文件</a></p>
      {% endblock %}
      </div>
    </div>


    <!-- Bootstrap core JavaScript
    ================================================== -->
    <!-- Placed at the end of the document so the pages load faster -->
    <script src="{% static './Bootstrap_files/jquery.min.js' %}"></script>
    <script>window.jQuery || document.write('<script
src="../../assets/js/vendor/jquery.min.js"><\/script>')</script>
    <script src="{% static './Bootstrap_files/bootstrap.min.js' %}"></script>
    <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
    <script src="{% static
'./Bootstrap_files/ie10-viewport-bug-workaround.js' %}"></script>

</body></html>
```

其余 html 继承 index，只需编辑<body>内容
例如 movie.html

```
{%extends "index.html" %}
{% block head %}
<li><a href="http://127.0.0.1:8000/index">首页</a></li>
<li class="active"><a href="http://127.0.0.1:8000/movies">豆瓣前 250 部电影</a></li>
<li><a href="http://127.0.0.1:8000/weathers">各地天气情况</a></li>
<li><a href="http://127.0.0.1:8000/phones">京东小米 9</a></li>
<li><a href="http://127.0.0.1:8000/updatabase">更新</a></li>
<form class="navbar-form navbar-right" action="/searchmovies" method="POST">
  {% csrf_token %}
    <input type="text" class="form-control" placeholder="{{searchkey}}" name="key">
    <button type="submit" class="btn btn-success">搜索</button>
</form>
{% endblock %}
{% block mainbody %}
<h1 >{{hello}}</h1>
<table class="table table-hover">
  <tr>
    <th >电影名称</th>
    <th >简介</th>
    <th >爬取时间</th>
    <th></th>
  </tr>
    {% for movie in movies %}
    <tr>
      <td><a href="{{movie.url}}">{{movie.name}}</a></td>
      <td>{{movie.synopsis}}</td>
      <td>{{movie.time}}</td>
      <td ><a class="btn btn-primary btn-sm" href="{{movie.url}}" role="button">查
看</a></td>
    </tr>
    {% endfor %}
</table>
{% endblock %}
```

control.py 调用爬虫，实现数据更新

```
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['hello']='删除成功！'
    except:
        context['hello']='删除失败，请先写入数据！'
    return render(request, 'index.html', context)

def insertdata(request):
    context = {}
    try:
        getmovies.get_movies()
        time.sleep(2)
        getweathers.get_weather()
        time.sleep(2)
        getphones.get_phones('小米 9')
        time.sleep(2)
        context['hello']='插入成功！'
    except:
        context['hello']='插入失败！'
    return render(request, 'index.html', context)

def updatabase(request):
    context = {}
    deletelall(request)
    insertdata(request)
    context['hello']='更新成功！'
    return render(request, 'index.html', context)
```