

爬虫期末作业报告

目录

爬虫期末作业报告1

 一、项目需求3

 二、技术简介3

 三、项目实现：4

 1、静态页面爬虫程序（豆瓣电影 top250）4

 2、动态页面爬虫程序（京东手机）7

 3、新建 Django 项目来展示数据：9

 4、12306 自动验证登陆..... 12

 5、完整项目目录..... 14

一、项目需求

- 1、编写爬虫程序爬取网页数据并存入数据库，再将数据利用 Django 展示在网页上。
- 2、实现 12306 自动验证登陆功能。

二、技术简介

爬虫：请求网站并提取数据的自动化程序。

Selenium：一个用于 Web 应用程序测试的工具。Selenium 测试直接运行在浏览器中，就像真正的用户在操作一样。支持的浏览器包括 IE)，[Mozilla Firefox](#), Safari, Google Chrome, Opera 等。这个工具的主要功能包括：测试与浏览器的兼容性——测试你的应用程序看是否能够很好得工作在不同浏览器和操作系统之上。测试系统功能——创建回归测试检验软件功能和用户需求。支持自动录制动作和自动生成 .Net、Java、Perl 等不同语言的测试脚本。

Mysql：最流行的关系型数据库管理系统。

Bootstrap：来自 Twitter，是目前最受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。

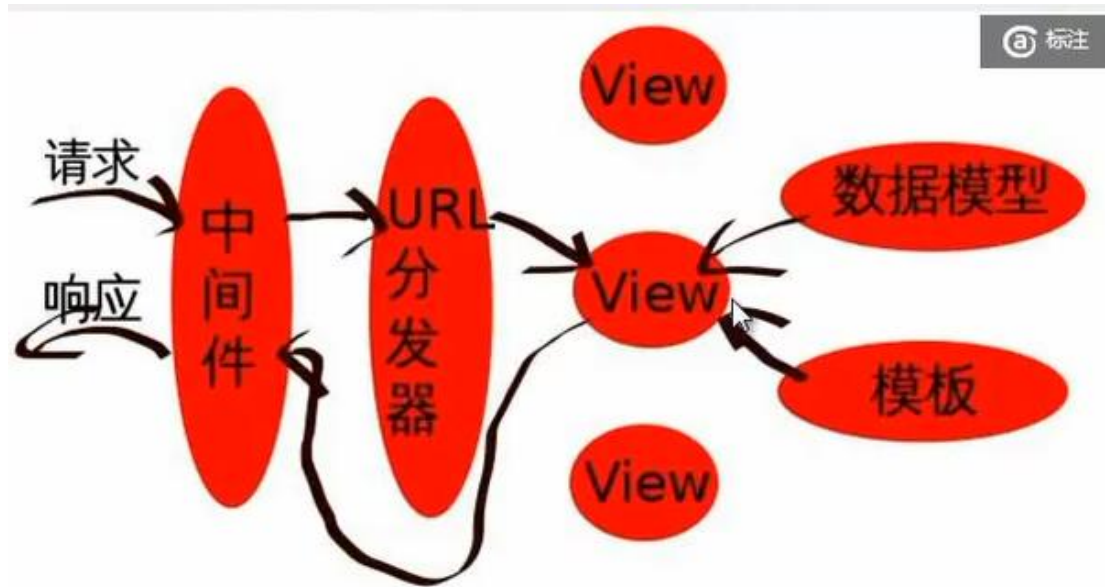
Django：一个开放源代码的 Web 应用框架，由 Python 写成。采用了 MVC 框架。适用于快速高效的上手开发一个网站。

Django MTV 模式：

m-models：用于业务对象和数据库之间的映射

t-templates：用于展示业务的模板页面

v-views：负责业务逻辑。



三、项目实施：

1、静态页面爬虫程序（豆瓣电影 top250）

1.1、实现思路：

用 request 库请求豆瓣页面，得到第一页的 html 代码；用正则表达式 re 库分析每条电影的正则，得到电影名、主演等信息放入字典。这样就得到第一页的 25 条电影信息；再点击下一页观察页面 url 的变化 <https://movie.douban.com/top250?start=2>，发现只有 start 的值改变，因此只需改变每次传入的 start 的值并循环调用 main (start) 方法 10 次即可。最后用多进程来提高爬取速度。

1.2、主要代码：

```
#请求网页
def get_one_page(url):
    try:
        response=requests.get(url)
        if response.status_code==200:
            return response.text
        return None
    except RequestException:
        return None
```

#解析单个页面

[illegible]

```
#存入 mysql
```

```
def write_to_mysql(data):
    db = pymysql.connect(host='localhost', user='root',
password='123456', db='myblog', charset='utf8')
    cur = db.cursor()
    sqlc = '''
        insert into mymovie_movies
        values(null,%s,%s,%s,%s,%s,%s,%s,%s)
    '''

    try:
        if cur.execute(sqlc, (data["title"], data["director"],
data["actor"], data["time"], data["region"], data["type"],
data["score"], data["images"])):
            print('Successful')
            db.commit()
    except Exception as e:
        print(e)
        print('Failed')
```

```

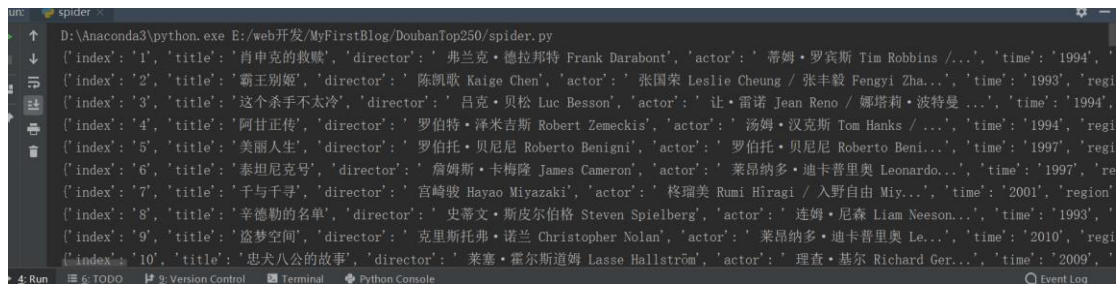
        db.rollback()
    cur.close()
    db.close()

def main(start):
    url = 'https://movie.douban.com/top250?start=' + str(start)
    html = get_one_page(url)
    for item in parse_one_page(html):
        print(item)

if __name__ == '__main__':
    pool = Pool()
    pool.map(main, [i*25 for i in range(10)])

```

1.3、效果截图;



开始事务		文本		筛选	排序	导入		导出	
index	title	director	actor	time	region	type	score	images	
1	肖申克	弗兰克·德拉邦特 Frank	蒂姆·罗宾斯 Tim Robbins / ...	1994	美国	犯罪 剧情	9.6	https://im	
2	霸王别姬	陈凯歌 Kaige Chen	张国荣 Leslie Cheung / 张丰毅 Fei	1993	中国大陆	剧情 爱情	9.6	https://im	
3	这个杀手	吕克·贝松 Luc Besson	让·雷诺 Jean Reno / 娜塔莉·波特	1994	法国	剧情 动作	9.4	https://im	
4	阿甘正传	罗伯特·泽米吉斯 Rober	汤姆·汉克斯 Tom Hanks / ...	1994	美国	剧情 爱情	9.4	https://im	
5	美丽人生	罗伯托·贝尼尼 Roberto	罗伯托·贝尼尼 Roberto Beni...	1997	意大利	剧情 喜剧	9.5	https://im	
6	泰坦尼克	詹姆斯·卡梅隆 James C	莱昂纳多·迪卡普里奥 Leonardo...	1997	美国	剧情 爱情	9.3	https://im	
7	千与千寻	宫崎骏 Hayao Miyazaki	柊瑠美 Rumi Hiragi / 入野自由 M	2001	日本	剧情 动画	9.3	https://im	
8	辛德勒的	史蒂文·斯皮尔伯格 Stev	连姆·尼森 Liam Neeson...	1993	美国	剧情 历史	9.5	https://im	
9	盗梦空间	克里斯托弗·诺兰 Christ	莱昂纳多·迪卡普里奥 Le...	2010	美国 英国	剧情 科幻	9.3	https://im	
10	忠犬八公	莱塞·霍尔斯道姆 Lasse	理查·基尔 Richard Ger...	2009	美国 英国	剧情	9.3	https://im	
11	机器人总	安德鲁·斯坦顿 Andrew	本·贝尔特 Ben Burtt / 艾丽...	2008	美国	爱情 科幻	9.3	https://im	
12	三傻大闹	拉库马·希拉尼 Rajkum	阿米尔·汗 Aamir Khan / 卡...	2009	印度	剧情 喜剧	9.2	https://im	
13	海上钢琴	朱塞佩·托纳多雷 Giusep	蒂姆·罗斯 Tim Roth / ...	1998	意大利	剧情 音乐	9.2	https://im	
14	放牛班的	克里斯托夫·巴拉蒂 Chr	热拉尔·朱尼奥 Gé...	2004	法国 瑞士	剧情 音乐	9.3	https://im	

2、动态页面爬虫程序（京东手机）

2.1、实现思路：

用 selenium 库模拟浏览器来爬取京东手机商品信息。首先下载对应的 chromedriver，并设置为无头浏览；通过 css 选择器来找到商品的输入框并输入“手机”且提交；同样的模拟翻页的情况，总共有 100 页；引入 pyquery 库，通过分析节点来获取到商品的不同信息；最后存入到数据库。

2.2、主要代码：

```
def search():
    try:
        browser.get("https://www.jd.com/")
        input = wait.until(EC.presence_of_element_located((By.ID,
'key'))))
        submit =
wait.until(EC.presence_of_element_located((By.CSS_SELECTOR,
'#search > div > div.form > button'))))
        input.send_keys(keyword)
        submit.click()
        total =
wait.until(EC.presence_of_element_located((By.CSS_SELECTOR,
'#J_bottomPage > span.p-skip > em:nth-child(1) > b'))))
        get_products()
        return total.text
    except TimeoutException:
        return search()

def next_page(page_number):
    print('正在翻页', page_number)
    try:
        # 滑动到底部，加载出后三十个货物信息
        browser.execute_script("window.scrollTo(0,
document.body.scrollHeight);")
        time.sleep(5)

        input =
wait.until(EC.presence_of_element_located((By.CSS_SELECTOR,
'#J_bottomPage > span.p-skip > input'))))
        submit =
wait.until(EC.element_to_be_clickable((By.CSS_SELECTOR,
```

```

' #J_bottomPage > span.p-skip > a' )))
    input.clear()
    input.send_keys(page_number)
    submit.click()
    wait.until(EC.text_to_be_present_in_element((By.CSS_SELECTOR,
' #J_bottomPage > span.p-num > a.curr'), str(page_number)))
    get_products()
except TimeoutException:
    next_page(page_number)

def get_products():
    wait.until(EC.presence_of_element_located((By.CSS_SELECTOR,
' #J_goodsList > ul' )))
    html = browser.page_source
    doc = pq(html)
    items = doc(' #J_goodsList .gl-warp .gl-item .gl-i-wrap').items()
    images = browser.find_element_by_xpath('//div[@class="gl-i-
wrap"] /div[1] /a /img')
    #获取商品信息列表
    for item in items:
        product = {
            'name': re.search('.*?\n', item.find('.p-
name').text()).group(0)[: -1],
            'price': item.find('.p-price').text()[2:],
            'commit': re.sub('\n', '', item.find('.p-
commit').text()),
            'shop': item.find('.p-shop').text(),
            'icons': re.sub('\n', ' ', item.find('.p-icons').text()),
            'image': images.get_attribute('src')
        }
        print(product)
        write_to_mysql(product)

```


2.3、效果截图：

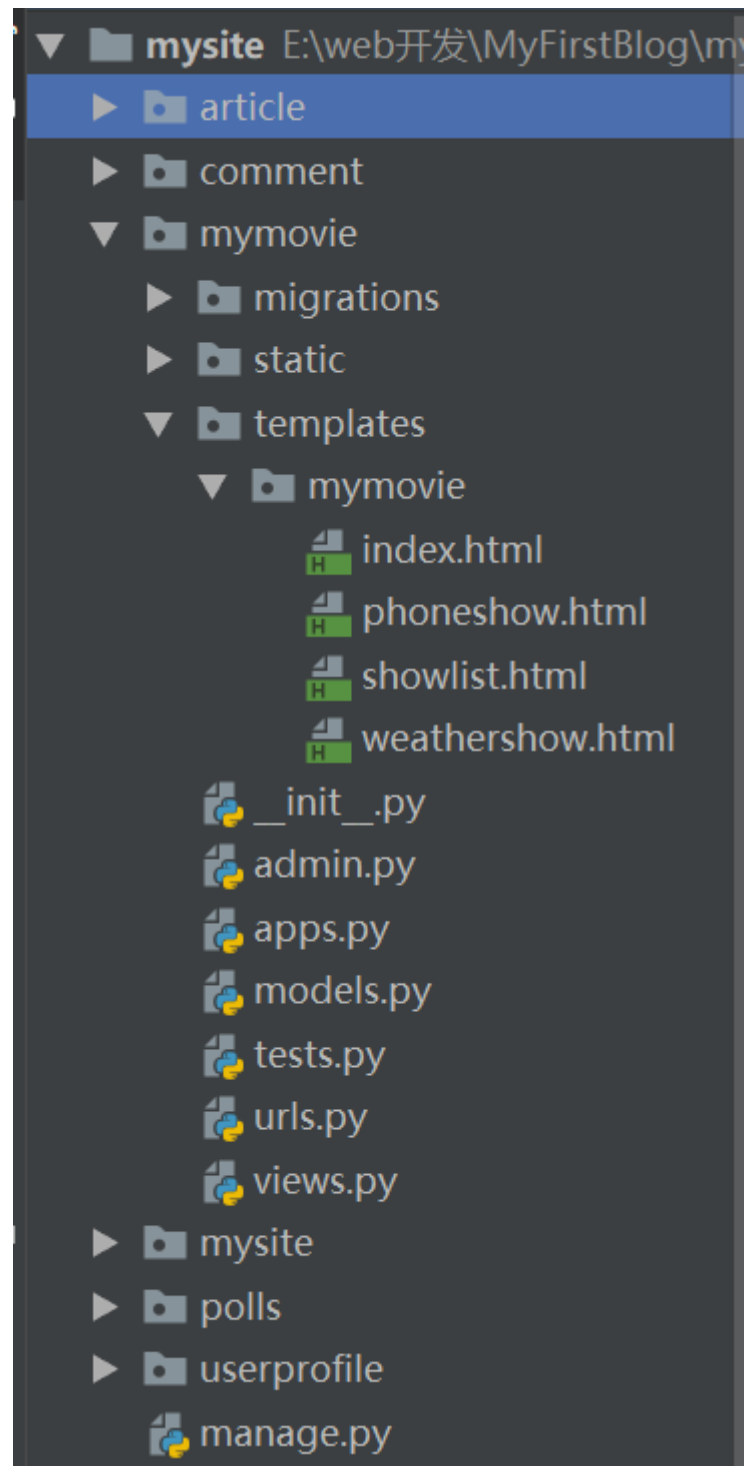
开始事务 文本 筛选 排序 导入 导出						
id	name	price	commit	shop	icons	image
1	OPPO Re	3599.00	1300+条评	OPPO京	自营 赠	https://img
2	Apple iPl	5899.00	二手有售91	Apple产	自营 券5	https://img
3	【KPL官	73298.00	二手有售8.4	vivo京东	自营 新品	https://img
4	荣耀8X 千	1298.00	二手有售14	荣耀京东	自营 放心	https://img
5	荣耀10青	1298.00	二手有售47	荣耀京东	自营 放心	https://img
6	vivo U1 2	799.00	13万+条评	vivo京东	自营 放心	https://img
7	小米 红米	1199.00	57万+条评	小米京东	自营 放心	https://img
8	荣耀V20	2798.00	二手有售23	荣耀京东	自营 满赠	https://img
9	OPPO Re	2999.00	0条评价	牧申手机	赠 险	https://img
10	荣耀畅玩	898.00	40万+条评	荣耀京东	自营 放心	https://img
11	小米 红米	799.00	二手有售70	小米京东	自营 放心	https://img

3、新建 Django 项目来展示数据：

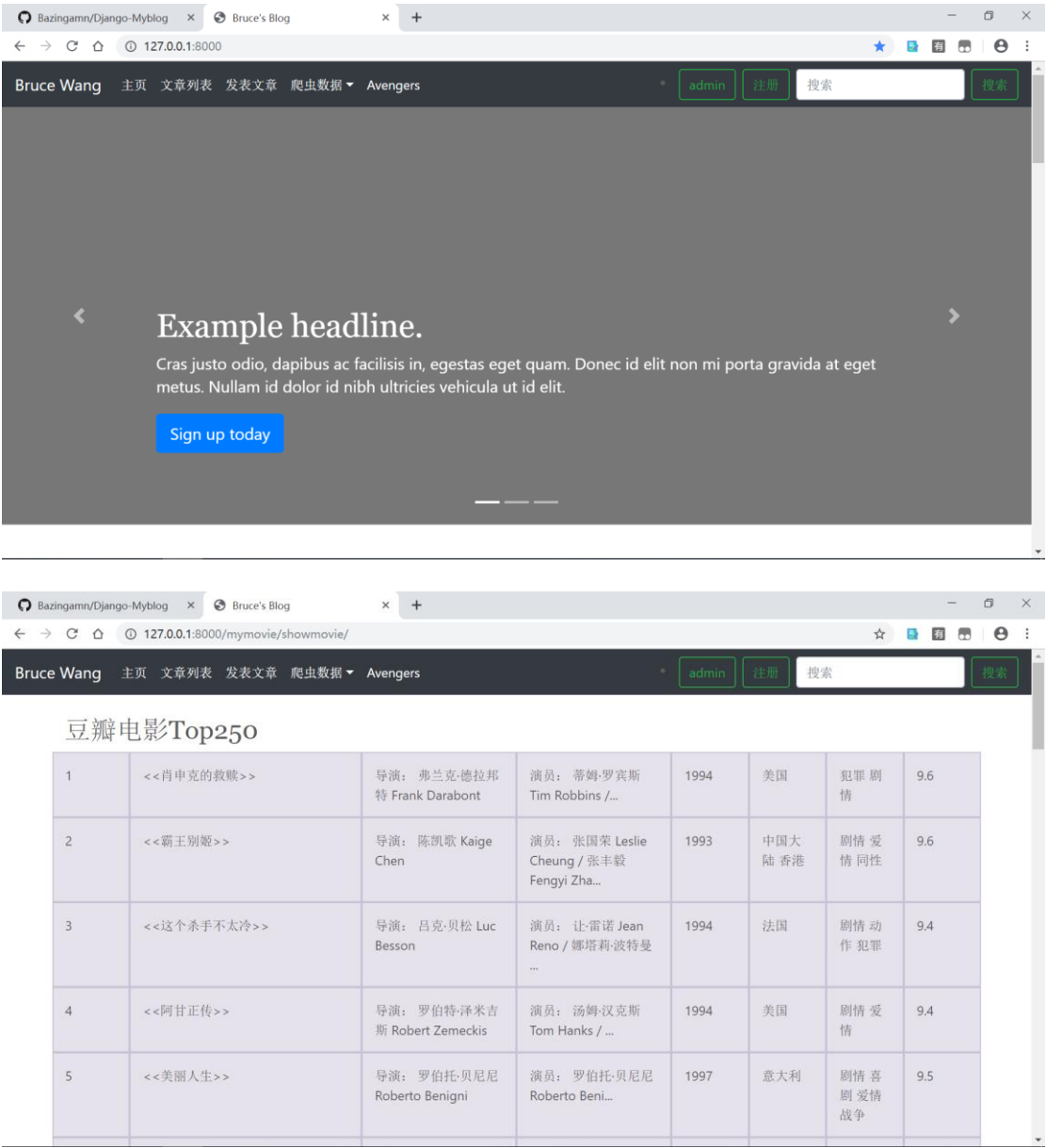
3.1、实现步骤：

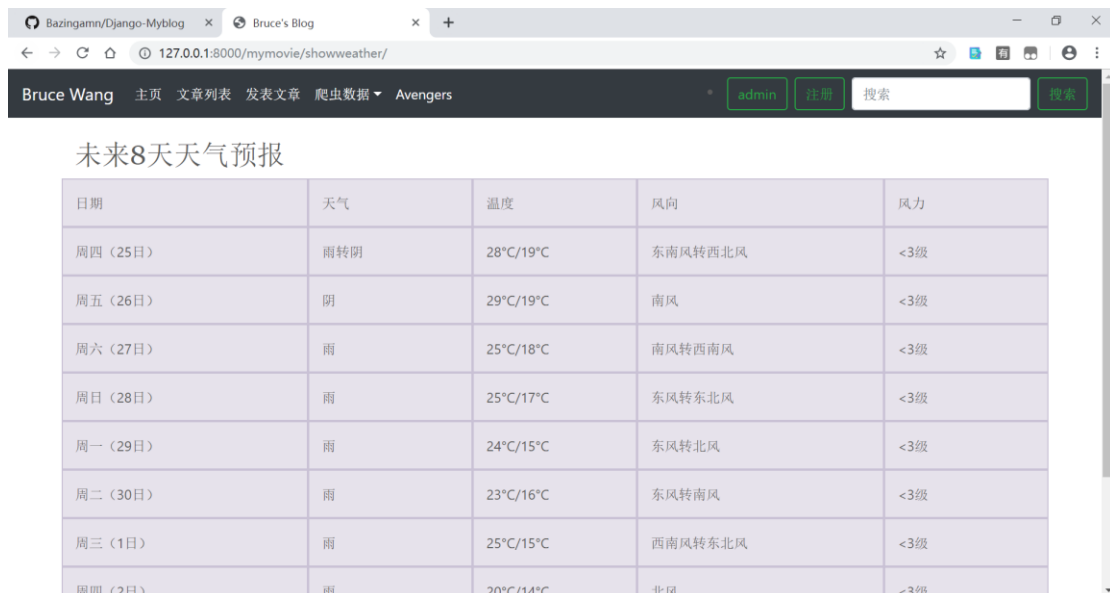
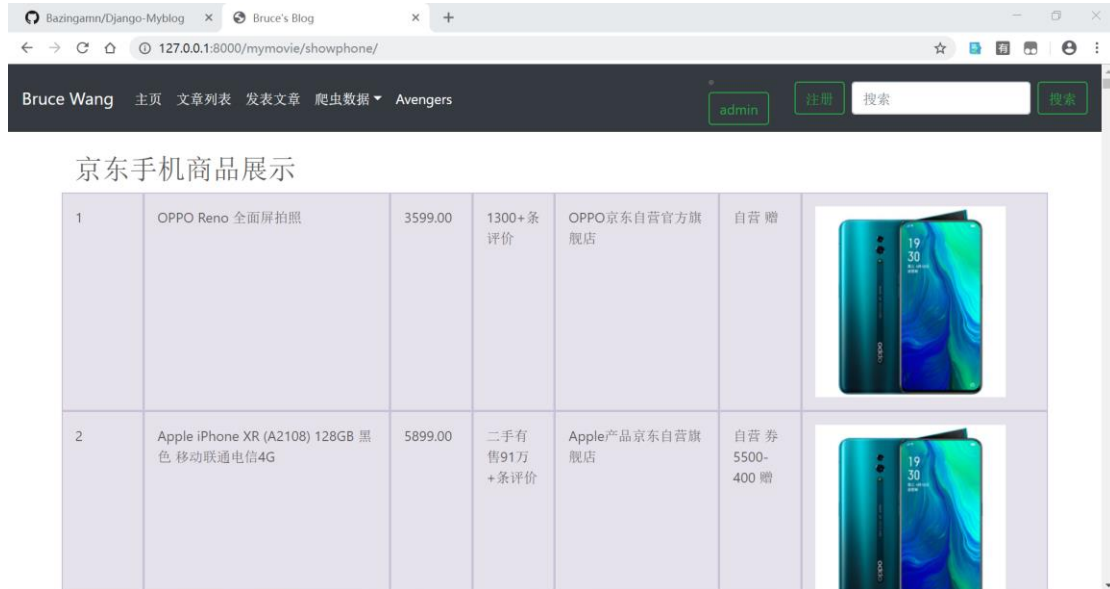
首先创建 models，映射到 mysql；创建 templates，编写要展示的 html 页面；在 views 里编写功能并调用 templates；编写 urls。

3.2、项目目录结构：



3.3、效果截图：





4、12306 自动验证登陆

```
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(u"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.coordina
te[i][0],

self.coordinate[i][1]).click()
        Action.perform()
    except Exception as e:
        print(e.message())


```


5、完整项目目录

 .idea

 12306verify

 DoubanTop250

 JDPhones

 WeatherForecast

 mysite

DouanTop250 为静态页面爬虫程序，JDPhones 为动态页面爬虫程序，mysite 为 Django 项目。