温州大学瓯江学院

爬虫 实验报告

实验名称：					
班　　级：	计算机三班	姓　　名：	潘以超	学　　号：	16219111325
实验地点：		日　　期：			

一、实验目的：

二、实验环境：

三、实验内容和要求：

四、实验步骤：

（对实验步骤的说明应该能够保证根据该说明即可重复完整的实验内容，得到正确结果。）

五、实验结果与分析（含程序、数据记录及分析和实验总结等）：

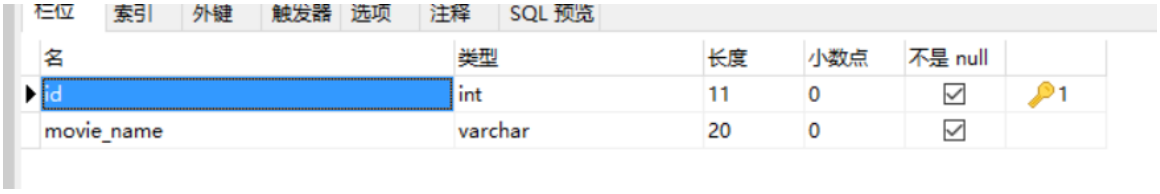
豆瓣 250：

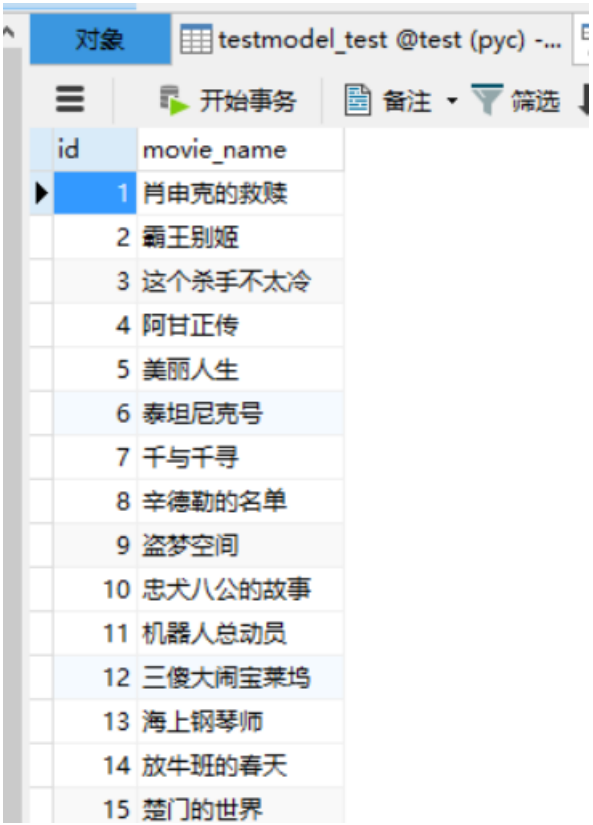
爬虫代码：

```
import requests
import lxml
import csv
import pymysql
from lxml import etree


def get_page():
    result = []
    for a in range(0, 10):
        url = 'https://movie.douban.com/top250?start=%s&filter=' % a*25
        res = requests.get(url)
        tree = etree.HTML(res.text)
```

```
        top250 = tree.xpath('//span[@class="title"][1]/text()')
        result += top250
    return result
res = get_page()
print(res)

db = pymysql.connect("localhost", "root", "123456", "test")
cursor = db.cursor()
print(res)
sql = "INSERT INTO testmodel_test(movie_name) VALUES(%s)"
for a in res:
    cursor.execute(sql, (a))
    db.commit()
db.close()
```

数据库：





网站效果：

京东手机：

爬虫代码：

```
from selenium import webdriver
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
import selenium.common.exceptions
import json
import csv
import time
import pymysql

class JdSpider():
    def open_file(self):
        self.fm = input('请输入文件保存格式（txt、json、csv）：')
```

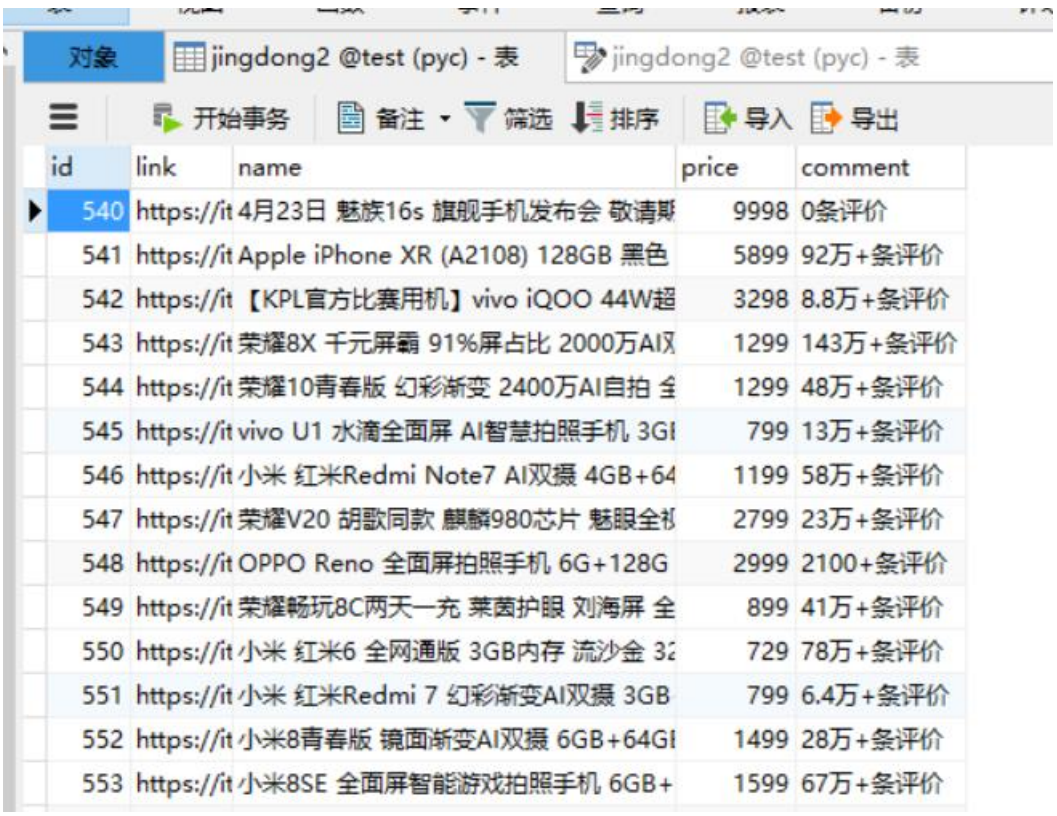
```
        while self.fm!='txt' and self.fm!='json' and self.fm!='csv':
            self.fm = input('输入错误，请重新输入文件保存格式（txt、json、csv）：')
        if self.fm=='txt' :
            self.fd = open('E:\爬虫\Jd.txt','w',encoding='utf-8')
        elif self.fm=='json' :
            self.fd = open('E:\爬虫\Jd.json','w',encoding='utf-8')
        elif self.fm=='csv' :
            self.fd = open('E:\爬虫\Jd.csv','w',encoding='utf-8',newline='')

    def open_browser(self):
        self.browser = webdriver.Firefox()
        self.browser.implicitly_wait(10)
        self.wait = WebDriverWait(self.browser,10)

    def init_variable(self):
        self.data = zip()
        self.isLast = False

    def parse_page(self):
        try:
            skus = self.wait.until(EC.presence_of_all_elements_located((By.XPATH,
                   '//li[@class="gl-item"]')))
            skus = [item.get_attribute('data-sku') for item in skus]
            links = ['https://item.jd.com/{sku}.html'.format(sku=item) for item in skus]
            prices = self.wait.until(EC.presence_of_all_elements_located((By.XPATH,
                     '//div[@class="gl-i-wrap"]/div[3]/strong/i')))
            prices = [item.text for item in prices]
            names = self.wait.until(EC.presence_of_all_elements_located((By.XPATH,
                   '//div[@class="gl-i-wrap"]/div[4]/a')))
            names = [item.text for item in names]
            comments = self.wait.until(EC.presence_of_all_elements_located((By.XPATH,
                       '//div[@class="gl-i-wrap"]/div[5]/strong')))
            comments = [item.text for item in comments]
            self.data = zip(links,prices,names,comments)
        except selenium.common.exceptions.TimeoutException:
            print('parse_page: TimeoutException')
            self.parse_page()
        except selenium.common.exceptions.StaleElementReferenceException:
            print('parse_page: StaleElementReferenceException')
            self.browser.refresh()

    def turn_page(self):
        try:

self.wait.until(EC.element_to_be_clickable((By.XPATH,'//a[@class="pn-next"]'))).click()
            time.sleep(1)
```

```
            self.browser.execute_script("window.scrollTo(0,document.body.scrollHeight)")
            time.sleep(2)
        except selenium.common.exceptions.NoSuchElementException:
            self.isLast = True
        except selenium.common.exceptions.TimeoutException:
            print('turn_page: TimeoutException')
            self.turn_page()
        except selenium.common.exceptions.StaleElementReferenceException:
            print('turn_page: StaleElementReferenceException')
            self.browser.refresh()

    def write_to_file(self):
        if self.fm == 'txt':
            for item in self.data:
                self.fd.write('----------------------------------------\n')
                self.fd.write('link: ' + str(item[0]) + '\n')
                self.fd.write('price: ' + str(item[1]) + '\n')
                self.fd.write('name: ' + str(item[2]) + '\n')
                self.fd.write('comment: ' + str(item[3]) + '\n')
        if self.fm == 'json':
            temp = ('link','price','name','comment')
            for item in self.data:
                json.dump(dict(zip(temp,item)),self.fd,ensure_ascii=False)
        if self.fm == 'csv':
            writer = csv.writer(self.fd)
            for item in self.data:
                writer.writerow(item)

    #def close_file(self):
        #self.fd.close()

    def close_browser(self):
        self.browser.quit()

    def crawl(self):
        #self.open_file()
        self.open_browser()
        self.init_variable()
        print('开始爬取')
        self.browser.get('https://search.jd.com/Search?keyword=
             %E6%89%8B%E6%9C%BA&enc=utf-8&pvid=ba6714c00a9a404a98475700b49df2f2')
        time.sleep(1)
        self.browser.execute_script("window.scrollTo(0,document.body.scrollHeight)")
        time.sleep(2)
        count = 0
        while not count==5:
```

```
            count += 1
            print('正在爬取第 ' + str(count) + ' 页......')
            self.parse_page()
            #self.write_to_file()
            self.write_to_mysql()
            self.turn_page()
        #self.close_file()
        self.close_browser()
        print('结束爬取')

    def write_to_mysql(self):
        db = pymysql.connect("localhost", "root", "123456", "test")
        cursor = db.cursor()
        sql = "INSERT INTO testmodel_jd(link,price,name,comment)VALUES(%s,%s,%s,%s)"
        for item in self.data:
            cursor.execute(sql, (item[0],item[1],item[2],item[3]))
            db.commit()
        db.close()

if __name__ == '__main__':
    spider = JdSpider()
    spider.crawl()
```

数据库：

名	类型	长度	小数点	不是 null	
id	int	20	0	☑	1
link	varchar	255	0	☐	
name	varchar	255	0	☐	
price	float	10	2	☐	
comment	varchar	255	0	☐	



对象 jingdong2 @test (pyc) - 表 jingdong2 @test (pyc) - 表

开始事务 备注 筛选 排序 导入 导出

id	link	name	price	comment
540	https://it	4月23日 魅族16s 旗舰手机发布会 敬请期	9998	0条评价
541	https://it	Apple iPhone XR (A2108) 128GB 黑色	5899	92万+条评价
542	https://it	【KPL官方比赛用机】vivo iQOO 44W超	3298	8.8万+条评价
543	https://it	荣耀8X 千元屏霸 91%屏占比 2000万AI双	1299	143万+条评价
544	https://it	荣耀10青春版 幻彩渐变 2400万AI自拍 全	1299	48万+条评价
545	https://it	vivo U1 水滴全面屏 AI智慧拍照手机 3GB	799	13万+条评价
546	https://it	小米 红米Redmi Note7 AI双摄 4GB+64	1199	58万+条评价
547	https://it	荣耀V20 胡歌同款 麒麟980芯片 魅眼全视	2799	23万+条评价
548	https://it	OPPO Reno 全面屏拍照手机 6G+128G	2999	2100+条评价
549	https://it	荣耀畅玩8C两天一充 莱茵护眼 刘海屏 全	899	41万+条评价
550	https://it	小米 红米6 全网通版 3GB内存 流沙金 32	729	78万+条评价
551	https://it	小米 红米Redmi 7 幻彩渐变AI双摄 3GB	799	6.4万+条评价
552	https://it	小米8青春版 镜面渐变AI双摄 6GB+64GB	1499	28万+条评价
553	https://it	小米8SE 全面屏智能游戏拍照手机 6GB+	1599	67万+条评价

网站效果图：

127.0.0.1:8000

https://item.jd.com/100005087998.html 9998.00 4月23日 魅族16s 旗舰手机发布会 敬请期待 0条评价

https://item.jd.com/100000177760.html 5899.00 Apple iPhone XR (A2108) 128GB 黑色 移动联通电信4G手机 双卡双待 【限时特价，抢券再减400元！选换修无忧版享抢！经典7系列！ 92万+条评价

https://item.jd.com/100003433872.html 3298.00 【KPL官方比赛用机】vivo iQOO 44W超快闪充 8GB+128GB电光蓝 全面屏拍照手机 骁龙855电竞游戏 全网通4G 8.8万+条

https://item.jd.com/8735304.html 1299.00 荣耀8X 千元屏霸 91%屏占比 2000万AI双摄 4GB+64GB 幻夜黑 移动联通电信4G全面屏 双卡双待 143万+条评价

https://item.jd.com/100001550349.html 1299.00 荣耀10青春版 幻彩渐变 2400万AI自拍 全网通版4GB+64GB 渐变蓝 移动联通电信4G全面屏 双卡双待 48万+条评价

https://item.jd.com/100002852992.html 1199.00 小米 红米Redmi Note7 AI双摄 4GB+64GB 亮黑色 全网通4G 双卡双待 水滴全面屏拍照游戏智能 58万+条评价

https://item.jd.com/100003332220.html 799.00 vivo U1 水滴全面屏 AI智慧拍照手机 3GB+32GB 极光色 移动联通电信全网通4G 13万+条评价

https://item.jd.com/100002544828.html 2799.00 荣耀V20 胡歌同款 麒麟980芯片 魅眼全视屏 4800万深感相机 6GB+128GB 幻夜黑 移动联通电信4G全面屏 23万+条评价

https://item.jd.com/100004404944.html 5488.00 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 3万+条评价

https://item.jd.com/100000650837.html 899.00 荣耀畅玩8C两天一充 莱茵护眼 刘海屏 全网通版4GB+32GB 幻夜黑 移动联通电信4G全面屏 双卡双待 41万+条评价

https://item.jd.com/7651903.html 729.00 小米 红米6 全网通版 3GB内存 流沙金 32GB 移动联通电信4G手机 双卡双待 78万+条评价

Setting.py:

```
"""
Django settings for HelloWorld project.

Generated by 'django-admin startproject' using Django 2.2.

For more information on this file, see
https://docs.djangoproject.com/en/2.2/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/2.2/ref/settings/
```

```
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'yy$oayqa+)z(!_578(+i=2$mk0mp$vnin)+2e(hwkfwuatoaya'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'TestModel',              # 添加此项
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'HelloWorld.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
```

```
        'DIRS': [BASE_DIR+"/templates",],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'HelloWorld.wsgi.application'


# Database
# https://docs.djangoproject.com/en/2.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',  # 或者使用 mysql.connector.django
        'NAME': 'test',
        'USER': 'root',
        'PASSWORD': '123456',
        'HOST':'localhost',
        'PORT':'3306',
    }
}


# Password validation
# https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
```

```
    },
]


# Internationalization
# https://docs.djangoproject.com/en/2.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.2/howto/static-files/

STATIC_URL = '/static/'
```

Models.py:

#创建数据库

$python manage.py makemigrations TestModel　　#让 Django 知道我们的模型有一些变更
$python manage.py migrate TestModel　　#创建表结构

```
# models.py
from django.db import models

class Test(models.Model):
    movie_name = models.CharField(max_length=20)

class JD(models.Model):
    link = models.CharField(max_length=255)
    price = models.CharField(max_length=10)
    name = models.CharField(max_length=255)
    comment = models.CharField(max_length=255)
```

testdb.py:

#读取数据并输出

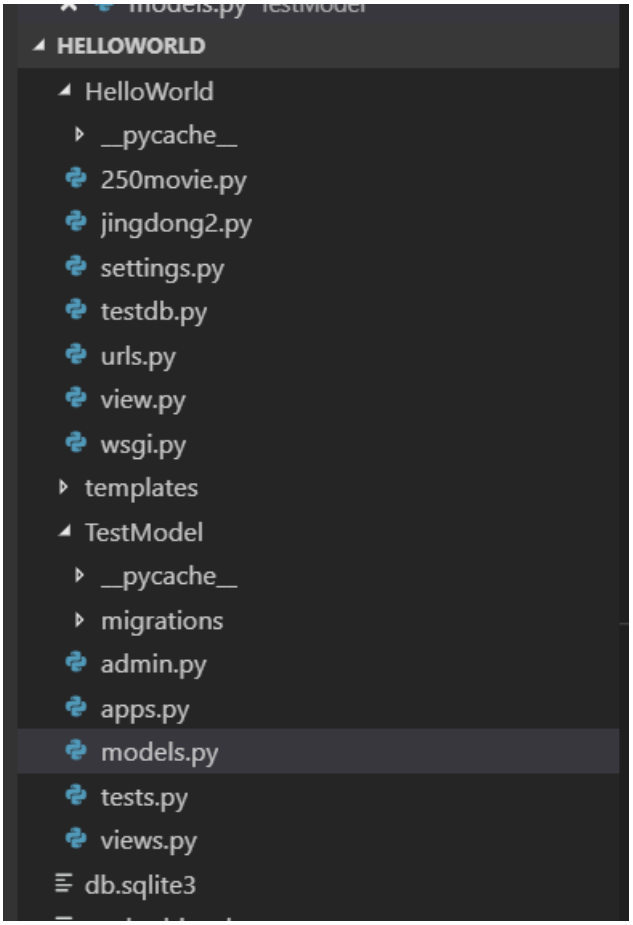
```
from django.http import HttpResponse

from TestModel.models import Test
from TestModel.models import JD
# 数据库操作
def testdb(request):
    response = ""
    response1 = ""
    list = Test.objects.all()
    for var in list:
        response1 +=str(var.id) + " "+var.movie_name + " " +"</p>"
    list2 = JD.objects.all()
    for var in list2:
        response1 += var.link + " " + var.price + " " + var.name + " " + var.comment + "
"+"</p>"
    response = response1
    return HttpResponse("<p>" + response + "</p>")
```

urls.py:

#注册页面并将数据传递到该页面

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

urlpatterns = [

    url(r'^$', testdb.testdb)
]
```

项目文件截图：



六：思考题：

七、教师评语：

实验成绩：　　教师：（签名要全称）　　年　月　日

注：1。此模板为专业实验报告的基本要求，若有特殊要求的实验，可在此模板基础上增加，但不可减少。
　　2．实验报告必须在学生提交报告后一星期内批改。

说明：

① 上下页边距改成 2 厘米，左边距为 2.0 厘米，右边距为 1.5 厘米。

② 表格位置为居中