**基于Python的当当网商品信息爬虫系统使用文档**

**1 前言**

1.1系统介绍

本系统是基于python语言设计的当当网商品信息爬虫系统，最终爬取到的数据保存在数据库中。

1.2 目标人群

通过阅读该操作手册，可以帮助想要学习python爬虫的人们尽快熟悉在windows系统中使用python开发进行爬虫的流程，并能成功进行爬虫设计。

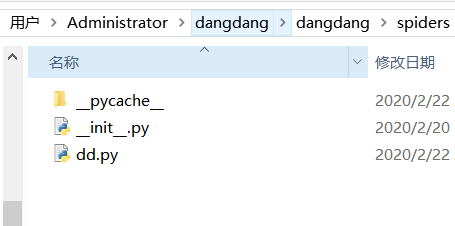
**2 使用说明**

2.1 创建爬虫项目

具体操作：打开cmd界面，用cd操作跳转到你想要存储文件的位置，输入：scrapy startproject dangdang，创建dangdang爬虫项目

2.2 创建一个以基础爬虫为模板的爬虫文件，命令为：dd.py

具体操作：在cmd界面输入：scrapy genspider -t dd dangdang.com即可。结果如下图所示，在spiders文件夹中生成了文件dd.py。



2.3 确定需要爬取的数据

具体操作：在配置文件items.py中创建要爬取的数据定义，具体代码如下：

import scrapy

class DangdangItem(scrapy.Item):

# define the fields for your item here like:

# name = scrapy.Field()

title = scrapy.Field()

link = scrapy.Field()

comment = scrapy.Field()

shop = scrapy.Field()

price = scrapy.Field()

2.4 确定网页以及数据的命名规律

具体操作：打开当当网（dangdang.com）,按F12查看网页源代码，确定想要爬虫的数据在网页中的命名方式。同时找寻多个网页地址的规律，便于编程。在爬虫文件dd.py中基于Xpath爬取网站信息，并且实现多页爬取。具体代码如下：

import sys

import os

curPath = os.path.abspath(os.path.dirname(\_\_file\_\_))

rootPath = os.path.split(curPath)[0]

sys.path.append(rootPath)

import scrapy

from dangdang.items import DangdangItem

from scrapy.http import Request # 依次爬取的工具

class DdSpider(scrapy.Spider):

name = 'dd'

allowed\_domains = ['dangdang.com']

start\_urls = ['http://category.dangdang.com/pgl-cid4008154.html']

header = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64:x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36',}

def parse(self, response):

item = DangdangItem()

item["title"] = response.xpath('//p[@name="title"]/a/@title').extract()

item["link"] = response.xpath('//p[@name="title"]/a/@href').extract()

item["comment"] = response.xpath('//a[@dd\_name="单品评论"]/text()').extract()

item["shop"] = response.xpath('//p/a[@dd\_name="单品店铺"]/text()').extract()

item["price"] = response.xpath('//span[@class="price\_n"]/text()').extract()

print(item)

yield item # 将数据提交到Pipeline中

for i in range(2,2001):

url = 'http://category.dangdang.com/pg'+str(i)+'-cid4008154.html'

# callback为回调函数

yield Request(url,callback=self.parse,headers=self.header)

2.5 爬虫风险规避

由于可能会出现爬虫失败，以及我们需要将获取的数据进行相关处理，因此我们在settings.py文件处，要修改三处：

① BOT\_NAME = 'dangdang

SPIDER\_MODULES = ['dangdang.spiders']

NEWSPIDER\_MODULE = 'dangdang.spiders'

② ROBOTSTXT\_OBEY = False

③ ITEM\_PIPELINES = { 'dangdang.pipelines.DangdangPipeline': 300,}

2.6 创建数据库和表

本系统选择的是MySQL数据库，采用Navicat for MySQL实现系统与数据库的简单连接。打开Navicat ，新建数据库，并且创建相应的SQL表，用于存储数据，命名为products。查询代码如下：

CREATE TABLE `products`(

`id` int(10) NOT NULL AUTO\_INCREMENT,

`shop` varchar(100) DEFAULT NULL COMMENT '出售店铺',

`title` varchar(100) DEFAULT NULL COMMENT '商品标题',

`link` varchar(100) DEFAULT NULL COMMENT '商品链接',

`price` varchar(100) DEFAULT NULL COMMENT '商品价格',

`comment` varchar(100) DEFAULT NULL COMMENT '商品评论',

PRIMARY KEY (`id`)

)ENGINE=INNODB DEFAULT CHARSET=utf8;

2.7 实现连接数据库功能

在Pipelines.py文件中编写相应的数据库链接函数，以及数据库存储操作将数据存入本地数据库中，具体代码如下：

import pymysql

def condb():

conn = pymysql.connect(host='localhost',port=3306,user='root',

password='cxks20121221!',db='dangdang',charset='utf8mb4')

return conn

class DangdangPipeline(object):

def process\_item(self, item, spider):

dbObject = condb()

cursor = dbObject.cursor()

sql = 'INSERT INTO products(shop,title,link,price,comment) VALUES(%s,%s,

%s,%s,%s)'

# 将数据导入到本地数据库中

for j in range(len(item['title'])):

try:

cursor.execute(sql,(item['shop'][j],item['title'][j],item['link'][j],

item['price'][j],item['comment'][j],))

cursor.close()

dbObject.commit()

except Exception as e:

print(e)

dbObject.rollback()

return item

2.8 运行系统功能

新建run.py，用于直接运行爬虫。具体代码如下：

from scrapy import cmdline

cmdline.execute('scrapy crawl zhihu\_topic'.split())

**3 注意事项**

第一，本系统开发环境：Windows10,python3.7,mysql,Navicat for MySQL

第二，本系统在不同电脑上使用时，只需将piplines.py中数据库服务器地址host、服务器端口号port、用户名user、密码password、数据库名称db、连接编码charset修改成与本机电脑中服务器一致，确保服务器能够连接成功，即可正确运行本程序。

第三，如果不希望出现scrapy日志文件，想要无日志输出，只需将语句：cmdline.execute('scrapy crawl dd'.split())修改为：cmdline.execute('scrapy crawl dd --nolog'.split())即可。

第四，可以通过修改dangdang文件夹中的dd.py中i的取值范围，来改变爬虫到的数据条数。