来来 scrapy 爬取各大网站每日热点新闻

作者：jackssybin
原文链接：https://ld246.com/article/1568913827437
来源网站：链滴

一.背景

最近玩爬虫，各种想爬，scrapy又非常好用。想多爬一点东西，决定爬一爬各大网站的热点新闻。
想到就开始做了哈
项目已经爬取：
豆瓣，　　微博，百度贴吧，虎扑，　github，百度今日热点

二.上代码

1.开始搭建项目

```
scrapy startproject crawl_everything
#起了个叼叼的名字
```

2.修改配置文件

- settings.py设置文件:

```
ROBOTSTXT_OBEY = False
# 下载延时
DOWNLOAD_DELAY = 0.5

#增加user-agent 这个可以拿自己浏览器的。也可以网上搜一些其他的。东西很多的
USER_AGENT = 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.54 Safari/536.5'
```

初步设想，我只需要存取文章的标题和内容链接和抓取时间

- 修改items.py

那么定义的item如下：

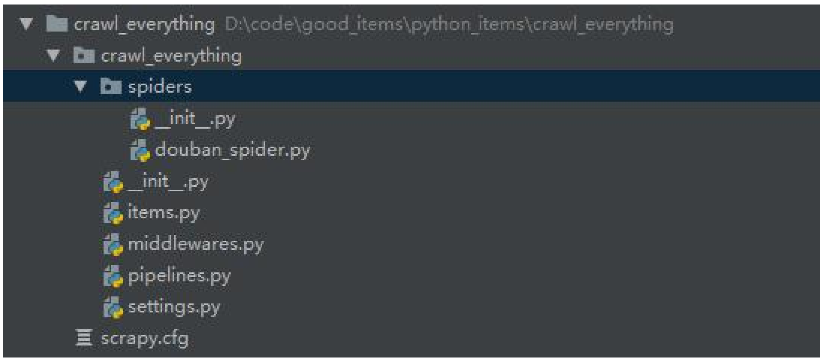
```
class CrawlEverythingItem(scrapy.Item):
    # 文章标题
    jk_title = scrapy.Field()
    # 文章链接
    jk_url = scrapy.Field()
    # 文章时间有可能没有
    jk_date = scrapy.Field()
```

3：生成第一个网站文件【豆瓣】：

#注意:这个命令在spiders目录执行

```
scrapy genspider douban_spider www.douban.com
```

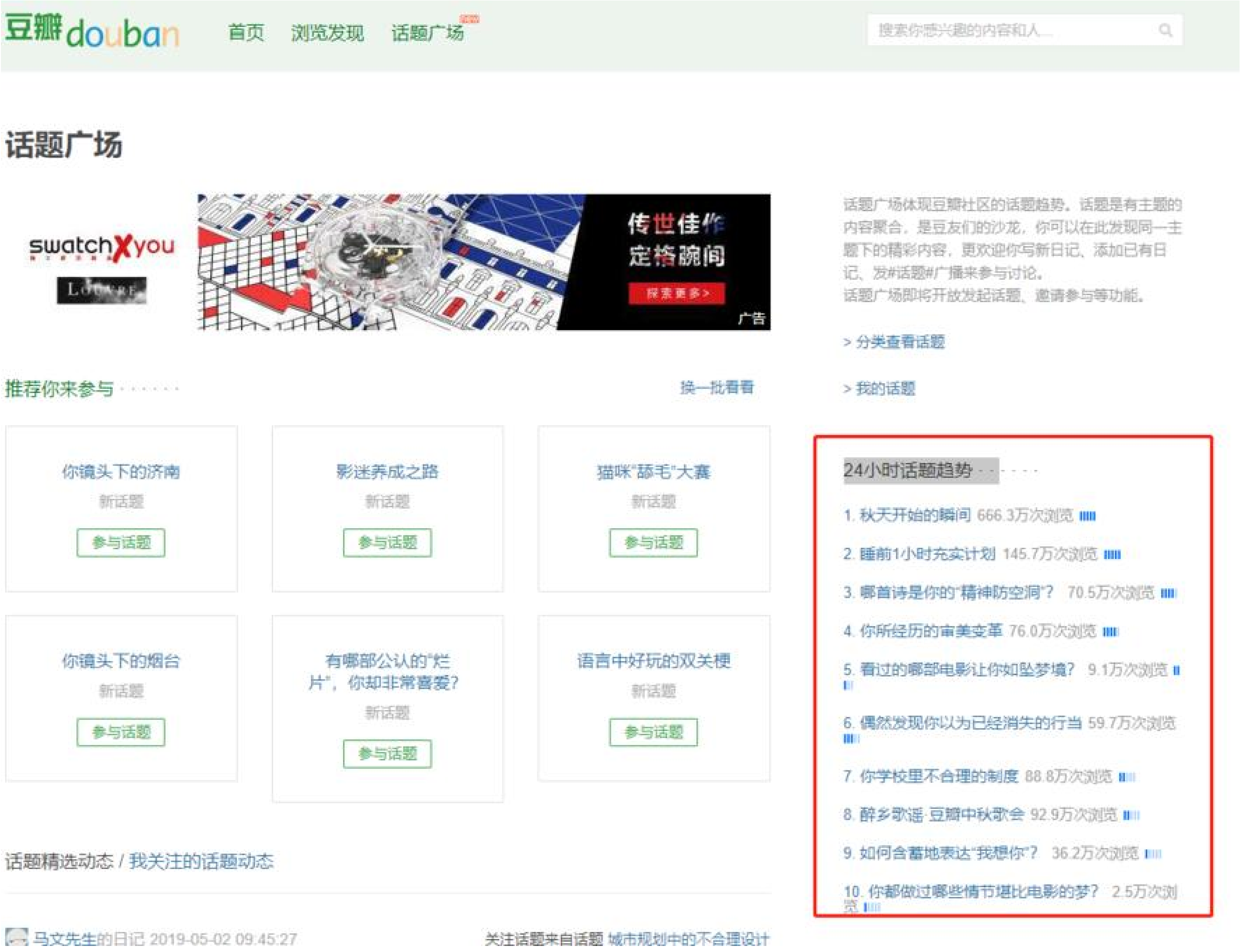
上个结果图：



```
▼ crawl_everything  D:\code\good_items\python_items\crawl_everything
  ▼ crawl_everything
    ▼ spiders
        __init__.py
        douban_spider.py
      __init__.py
      items.py
      middlewares.py
      pipelines.py
      settings.py
    scrapy.cfg
```

#先抓取豆瓣上的【24小时话题趋势】看看大家都在讨论啥

目标网址:https://www.douban.com/gallery/



话题广场

话题广场体现豆瓣社区的话题趋势。话题是有主题的内容聚合，是豆友们的沙龙，你可以在此发现同一主题下的精彩内容，更欢迎你写新日记、添加已有日记、发#话题#广播来参与讨论。
话题广场即将开放发起话题、邀请参与等功能。

24小时话题趋势

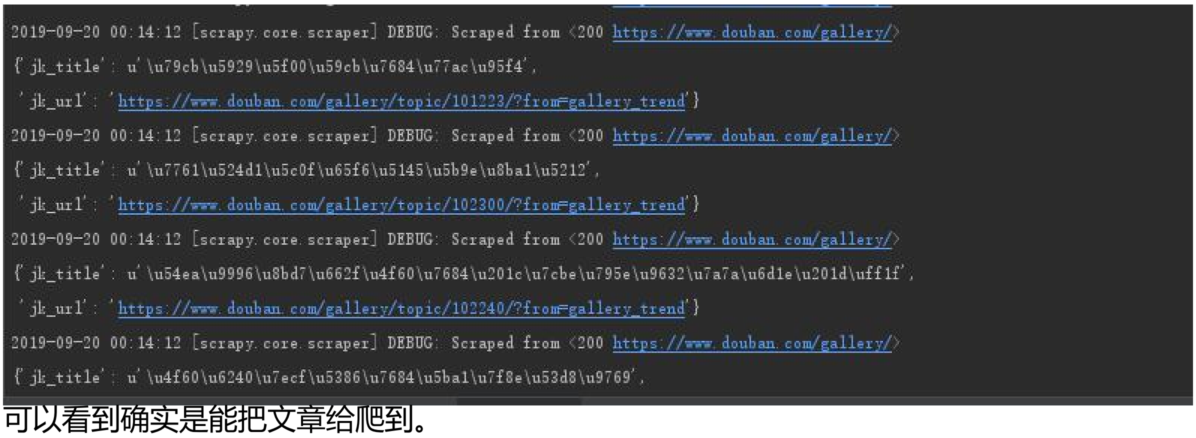
1. 秋天开始的瞬间 666.3万次浏览
2. 睡前1小时充实计划 145.7万次浏览
3. 哪首诗是你的"精神防空洞"？ 70.5万次浏览
4. 你所经历的审美变革 76.0万次浏览
5. 看过的哪部电影让你如坠梦境？ 9.1万次浏览
6. 偶然发现你以为已经消失的行当 59.7万次浏览
7. 你学校里不合理的制度 88.8万次浏览
8. 醉乡歌谣·豆瓣中秋歌会 92.9万次浏览
9. 如何含蓄地表达"我想你"？ 36.2万次浏览
10. 你都做过哪些情节堪比电影的梦？ 2.5万次浏览

知道目标了就开始爬吧：通过chrome的debug功能找到对应标签

```
 def parse(self, response):
        trends=response.css('ul.trend > li > a')
        for trend in trends:
            item = CrawlEverythingItem()
            item['jk_title']=trend.css('a::text').extract_first()
            item['jk_url'] = trend.css('a').attrib['href']
            yield item
```

程序这就写好了。运行一下

```
scrapy crawl douban_spider
```



```
2019-09-20 00:14:12 [scrapy.core.scraper] DEBUG: Scraped from <200 https://www.douban.com/gallery/>
{'jk_title': u'\u79cb\u5929\u5f00\u59cb\u7684\u77ac\u95f4',
 'jk_url': 'https://www.douban.com/gallery/topic/101223/?from=gallery_trend'}
2019-09-20 00:14:12 [scrapy.core.scraper] DEBUG: Scraped from <200 https://www.douban.com/gallery/>
{'jk_title': u'\u7761\u524d1\u5c0f\u65f6\u5145\u5b9e\u8ba1\u5212',
 'jk_url': 'https://www.douban.com/gallery/topic/102300/?from=gallery_trend'}
2019-09-20 00:14:12 [scrapy.core.scraper] DEBUG: Scraped from <200 https://www.douban.com/gallery/>
{'jk_title': u'\u54ea\u9996\u8bd7\u662f\u4f60\u7684\u201c\u7cbe\u795e\u9632\u7a7a\u6d1e\u201d\uff1f',
 'jk_url': 'https://www.douban.com/gallery/topic/102240/?from=gallery_trend'}
2019-09-20 00:14:12 [scrapy.core.scraper] DEBUG: Scraped from <200 https://www.douban.com/gallery/>
{'jk_title': u'\u4f60\u6240\u7ecf\u5386\u7684\u5ba1\u7f8e\u53d8\u9769',
```

可以看到确实是能把文章给爬到。

但每次需要输入命令颇为麻烦。

为了简单弄了个main函数主程序【mainCrawlEveryThing.py】，方便启动和debug

内容如下:

```
# coding: utf-8
from scrapy.cmdline import execute
import sys
import os

sys.path.append(os.path.dirname(os.path.abspath(__file__)))
execute(['scrapy', 'crawl', 'douban_spider'])  # 你需要将此处的douban_spider替换为你自己的爬名称
```

运行程序，效果和在cmd里输入命令效果一致。

4：数据入库:

接下来数据爬到了该入库了。

新建了一个数据库和一张表

```
create database crawl_everything;

drop TABLE `article_info`;
CREATE TABLE `article_info` (
  `article_id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '文章id',
  `jk_source` varchar(50) DEFAULT NULL COMMENT '文章来源',
  `jk_title` varchar(200) DEFAULT NULL COMMENT '文章标题',
  `jk_url` varchar(200) DEFAULT NULL COMMENT '文章url',
  `jk_status` varchar(50) DEFAULT '0' COMMENT '状态 0:禁用，1:正常',
  `jk_remark` varchar(500) DEFAULT NULL COMMENT '备注',
  `jk_create` datetime DEFAULT NULL COMMENT '创建时间',
  PRIMARY KEY (`article_id`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8 COMMENT='文章相关';
```

修改配置文件settings.py

```
ITEM_PIPELINES = {
   'crawl_everything.pipelines.CrawlEverythingPipeline': 400
}
```

修改配置文件pipelines.py

然后保存数据更改一下

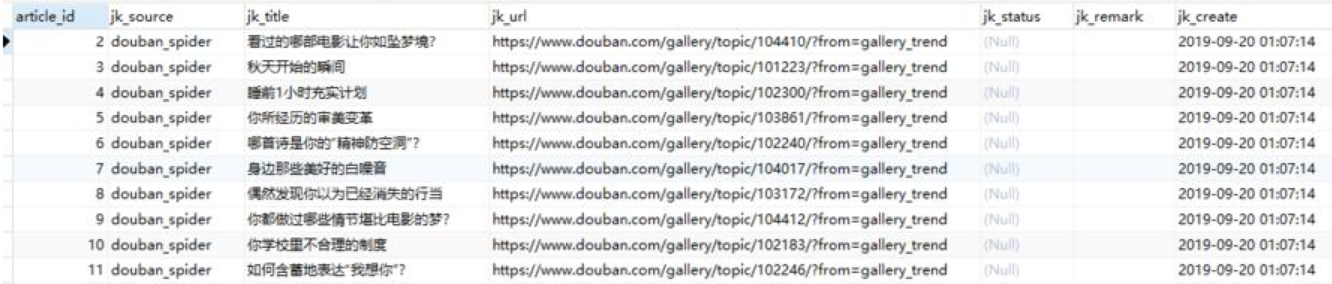
```
import  pymysql
import datetime;
import sys;
reload(sys);
sys.setdefaultencoding("utf8")

class CrawlEverythingPipeline(object):
    def __init__(self):
        # 连接MySQL数据库
        self.connect = pymysql.connect(host='localhost', user='root', password='root1234', db='
rawl_everything', port=3307)
        self.cursor = self.connect.cursor()

    def process_item(self, item, spider):
        # 往数据库里面写入数据
        self.cursor.execute(
            'INSERT INTO crawl_everything.article_info( jk_source, jk_title, jk_url, jk_remark, jk_crea
e) '
            'VALUES("{}","{}","{}","{}","{}")'.format(item['jk_source'], item['jk_title'],item['jk_url'],item['
k_remark'],datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S"))
        )
        self.connect.commit()
        return item
        # 关闭数据库

    def close_spider(self, spider):
        self.cursor.close()
        self.connect.close()
```

再次重新运行主函数，爬取到的数据已经入库



article_id	jk_source	jk_title	jk_url	jk_status	jk_remark	jk_create
2	douban_spider	看过的哪部电影让你如坠梦境?	https://www.douban.com/gallery/topic/104410/?from=gallery_trend	(Null)		2019-09-20 01:07:14
3	douban_spider	秋天开始的瞬间	https://www.douban.com/gallery/topic/101223/?from=gallery_trend	(Null)		2019-09-20 01:07:14
4	douban_spider	睡前1小时充实计划	https://www.douban.com/gallery/topic/102300/?from=gallery_trend	(Null)		2019-09-20 01:07:14
5	douban_spider	你所经历的审美变革	https://www.douban.com/gallery/topic/103861/?from=gallery_trend	(Null)		2019-09-20 01:07:14
6	douban_spider	哪首诗是你的"精神防空洞"?	https://www.douban.com/gallery/topic/102240/?from=gallery_trend	(Null)		2019-09-20 01:07:14
7	douban_spider	身边那些美好的白噪音	https://www.douban.com/gallery/topic/104017/?from=gallery_trend	(Null)		2019-09-20 01:07:14
8	douban_spider	偶然发现你以为已经消失的行当	https://www.douban.com/gallery/topic/103172/?from=gallery_trend	(Null)		2019-09-20 01:07:14
9	douban_spider	你都做过哪些情节堪比电影的梦?	https://www.douban.com/gallery/topic/104412/?from=gallery_trend	(Null)		2019-09-20 01:07:14
10	douban_spider	你学校里不合理的制度	https://www.douban.com/gallery/topic/102183/?from=gallery_trend	(Null)		2019-09-20 01:07:14
11	douban_spider	如何含蓄地表达"我想你"?	https://www.douban.com/gallery/topic/102246/?from=gallery_trend	(Null)		2019-09-20 01:07:14

后续在完善:先开门红一个豆瓣。逐步完善

用一个爬虫主类增加了几个网站的爬取，

```
 def parse(self, response):
        if response.url == self.start_urls[0]:#豆瓣
            yield from self.crawl_douban(response)
        if response.url == self.start_urls[1]:  # 微博
            yield from self.crawl_weibo(response)
        if response.url == self.start_urls[2]:  # 百度贴吧
            yield from self.crawl_tieba(response)
        if response.url == self.start_urls[3]:  # 虎扑
            yield from self.crawl_hupu(response)
        if response.url == self.start_urls[4]:  # github
            yield from self.crawl_github(response)
        if response.url == self.start_urls[5]:  # 百度今日热点
            yield from self.crawl_topbaidu(response)
    def crawl_douban(self, response):
        trends = response.css('ul.trend > li > a')
        for trend in trends:
            item = CrawlEverythingItem()
            item['jk_source'] = 'douban_spider'
            item['jk_title'] = trend.css('a::text').extract_first()
            item['jk_url'] = trend.css('a').attrib['href']
            item['jk_remark'] = ''
            yield item

    def crawl_weibo(self, response):
        trends = response.css('td.td-02 > a')
        for trend in trends:
            item = CrawlEverythingItem()
            item['jk_source'] = 'weibo_spider'
            item['jk_title'] = trend.css('a::text').extract_first()
            href = self.get_weibo_href(trend)
            item['jk_url'] = "https://s.weibo.com" + href
            item['jk_remark'] = ''
            yield item

    def crawl_tieba(self, response):
        trends = response.css('div.main > ul > li  a')
        for trend in trends:
            item = CrawlEverythingItem()
            item['jk_source'] = 'tieba_spider'
            item['jk_title'] = trend.css('a::text').extract_first()
            item['jk_url'] = trend.css('a').attrib['href']
            item['jk_remark'] = ''
            yield item

    def crawl_hupu(self, response):
        trends = response.css('div.list> ul > li >span:nth-child(1) >a')
        for trend in trends:
            item = CrawlEverythingItem()
            item['jk_source'] = 'hupu_spider'
            item['jk_title'] = trend.css('a').attrib['title']
            item['jk_url'] ="https://bbs.hupu.com"+trend.css('a').attrib['href']
            item['jk_remark'] = ''
            yield item

    def crawl_github(self, response):
        trends = response.css('div> article.Box-row ')
        for trend in trends:
            item = CrawlEverythingItem()
```

```
            item['jk_source'] = 'github_spider'
            jk_title="".join(trend.css('p::text').extract())
            re.sub(r'[\\*|  “<>:/()  （）  0123456789]', '', jk_title)
            jk_title.replace('\n', '').replace('　', '')
            item['jk_title'] = jk_title
            item['jk_url'] = "https://github.com" + trend.css('h1>a').attrib['href']
            item['jk_remark'] = ''
            yield item

    def crawl_topbaidu(self, response):
        trends = response.css('td.keyword >a:nth-child(1) ')
        for trend in trends:
            item = CrawlEverythingItem()
            item['jk_source'] = 'topbaidu_spider'
            item['jk_title'] = trend.css('a::text').extract_first()
            item['jk_url'] = trend.css('a').attrib['href']
            item['jk_remark'] = ''
            yield item

    def get_weibo_href(self,trend):
        href = trend.css('a').attrib['href']
        if href.startswith('javascript'):  ##javascript:void(0)
            href = trend.css('a').attrib['href_to']
        return href
```

增加了一个爬虫主类mainCrawlEveryThing.py

```
# coding: utf-8
from scrapy.cmdline import execute
import sys
import os
sys.path.append(os.path.dirname(os.path.abspath(__file__)))

execute(['scrapy', 'crawl', 'crawl_everything_spider'])  # 你需要将此处的douban_spider替换为你己的爬虫名称
```

数据库增加一个日期字段，修改了创建时间字段。

git已修改，并上传。

git已上传：https://github.com/jackssybin/crawl_everything