**新闻爬虫程序说明文档**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **日期** | **作者** | **描述** |
| v1.0.0 | 2022.05.10 | 王彦彦 |  |

本文档主要介绍新闻爬虫程序功能实现类及配置项具体参数，适用于程序部署指导及相关开发人员阅读。

# 一、功能描述

新闻程序主要功能是python的scrapy框架爬取新闻信息存储到oracle数据库中，通过搭建web服务器使用http请求展示在Qt界面上。

# 二、爬取网站

国务院 http://www.gov.cn/guowuyuan/index.htm 动态

银保监会 http://www.cbirc.gov.cn/cn/view/pages/index/index.html 监管动态

证监会 http://www.csrc.gov.cn/ 时政要闻 证监会要闻

央行 http://www.pbc.gov.cn/ 新闻

财政部 http://www.mof.gov.cn/index.htm 财政新闻 政策发布

发改委 https://www.ndrc.gov.cn/ 新闻发布|时政要闻|通知公告

国家统计局 http://www.stats.gov.cn/ 最新发布与解读

上海交易所 http://www.sse.com.cn/ 要闻 本所动态

深圳交易所 http://www.szse.cn/ 时政要闻 深交所要闻

中国证券报

上海证券报

证券时报

证券日报

# 三、依赖库



注：linux安装chromium

pacman -Sy chromium

# 四、实现

|  |
| --- |
| base\_url = 'http://www.csrc.gov.cn/csrc/c100027/common\_list.shtml' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_csrc\_newboard, encoding='utf-8',  meta={"\_id": 1}) base\_url = "http://www.csrc.gov.cn/searchList/a1a078ee0bc54721ab6b148884c784a8?\_isAgg=true&\_isJson=true&\_pageSize=18&\_template=index&\_rangeTimeGte=&\_channelName=&page=1" yield scrapy.Request(url=base\_url, method='GET', meta={"method": "json", "\_id": 2},  callback=self.parse\_csrc\_event,  encoding='utf-8') base\_url = 'http://www.gov.cn/guowuyuan/index.htm' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_gov\_newboard, encoding='utf-8',  meta={"\_id": 3}) base\_url = 'http://www.cbirc.gov.cn/cn/static/data/DocInfo/SelectItemAndDocByItemPId/data\_itemId=914,pageSize=10.json' yield scrapy.Request(url=base\_url, method='GET', meta={"method": "json", "\_id": 4},  callback=self.parse\_cbirc\_newboard,  encoding='utf-8') base\_url = 'http://www.pbc.gov.cn/goutongjiaoliu/113456/113469/index.html' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_pboc\_newboard, encoding='utf-8',  meta={"\_id": 5}) base\_url = 'http://www.mof.gov.cn/zhengwuxinxi/caizhengxinwen/' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_mof\_newboard, encoding='utf-8',  meta={"\_id": 6}) base\_url = 'http://www.mof.gov.cn/zhengwuxinxi/zhengcefabu/' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_mof\_newboard, encoding='utf-8',  meta={"\_id": 7}) base\_url = 'https://www.ndrc.gov.cn/xwdt/xwfb/?code=&state=123' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_ndrc\_xwfb\_newboard, encoding='utf-8',  meta={"\_id": 8}) base\_url = 'https://www.ndrc.gov.cn/xwdt/szyw/?code=&state=123' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_ndrc\_xwfb\_newboard, encoding='utf-8',  meta={"\_id": 9}) base\_url = 'http://www.stats.gov.cn/tjsj/zxfb/' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_stats\_newboard, encoding='utf-8',  meta={"\_id": 10}) base\_url = 'http://www.sse.com.cn/home/component/news/' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_sse\_newboard, encoding='utf-8',  meta={"\_id": 11}) base\_url = 'http://www.sse.com.cn/aboutus/mediacenter/hotandd/' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_sse\_newboard, encoding='utf-8',  meta={"\_id": 12}) base\_url = 'http://www.szse.cn/index/news/index.html' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_szse\_ss\_newboard, encoding='utf-8',  meta={"\_id": 13}) base\_url = 'http://www.szse.cn/aboutus/trends/news/' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_szse\_bs\_newboard, encoding='utf-8',  meta={"\_id": 14}) base\_url = 'https://www.stcn.com/xw/' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_stcn\_xw\_newboard, encoding='utf-8',  meta={"\_id": 15}) base\_url = 'http://www.stcn.com/kuaixun/' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_stcn\_kx\_newboard, encoding='utf-8',  meta={"\_id": 16}) base\_url = 'http://www.zqrb.cn/finance/hongguanjingji/index.html' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_zqrb\_finance\_newboard, encoding='utf-8',  meta={"\_id": 17}) base\_url = 'http://www.zqrb.cn/stock/gupiaoyaowen/index.html' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_zqrb\_stock\_newboard, encoding='utf-8',  meta={"\_id": 18}) base\_url = 'http://www.zqrb.cn/stock/redian/index.html' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_zqrb\_stock\_red\_newboard, encoding='utf-8',  meta={"\_id": 19}) base\_url = 'https://www.cs.com.cn/xwzx/hg/' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_cs\_hg\_newboard, encoding='utf-8',  meta={"\_id": 20}) base\_url = 'https://news.cnstock.com/news/sns\_yw/index.html' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_cnstock\_news\_yw\_newboard, encoding='utf-8',  meta={"\_id": 21}) base\_url = 'https://news.cnstock.com/news/sns\_jg/index.html' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_cnstock\_news\_jg\_newboard, encoding='utf-8',  meta={"\_id": 22}) base\_url = 'http://www.news.cn/politicspro/' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_news\_politicspro\_newboard,  encoding='utf-8', meta={"\_id": 23}) base\_url = 'http://www.news.cn/fortunepro/' yield scrapy.Request(url=base\_url, method='GET', callback=self.parse\_news\_fortunepro\_newboard,  encoding='utf-8', meta={"\_id": 24}) |
| 功能：爬取的各个网站 |
| class ChromeSeleniumDownloaderMiddlewareOne;  功能：ChromeSelenium中间件 |
| class CsrcPipeline;  功能：管道将数据写入oracle |
| ORACLE\_USER = 'bjhy' ORACLE\_PASSWORD = 'bjhy' ORACLE\_HOST = '192.168.101.215' ORACLE\_PORT = 1521 ORACLE\_SERVICE\_NAME = 'bjhy' ORACLE\_TABLE = "t\_o\_csrc\_official\_news"  current\_day = datetime.datetime.now() # LOG\_ENABLED = True LOG\_ENCODING = 'utf-8' LOG\_FILE = "./log/{}-{}-{}.log".format(current\_day.year, current\_day.month, current\_day.day)  功能：数据库和日志配置。 |
| @scheduler.scheduled\_job("cron", day\_of\_week="0-4", minute="\*/1", hour="9-15", id='task', max\_instances=100) def job():  os.system("scrapy crawl csrc\_news\_info ")  功能：定时任务 |
|  |

# 五、中间件

oracle、ChromeSelenium

# 六、源码

<http://192.168.101.210:7990/scm/~wangyanyan/kafka-stock-repo.git>

# 七、部署

测试环境：

启动爬虫程序：./run.sh 停止：stop.sh

启动web服务：./server.sh 停止：server\_stop.sh

日志：./log