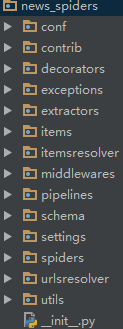
**关于新闻项目的开发文档**

该项目是基于Python Scrapy 框架定制， 主要解决为京东，智投App提供财经类新闻源资讯， 要理解改项目的相关开发， 需先理解Scrapy和相关流程， 下面是关于Scrapy的执行流程，以作参考：



**一、下面对该项目的执行的相关模块进行说明：**

1：基于Scrapy框架的该项目目录(news\_spiders)如下：



2：spider/, 包括common和news\_spiders模块

common 模块：定制基本的spider类， 起始链接， 各网站配置加载。

news\_spiders模块：提供解析新闻各个字段， 主要是基于parser\_news方法。

3：urlsresolver/， 主要pagelinks 和 resolver模块

pagelinks 模块：抓取一篇新闻时可能会有多页， 提取并规范化相应链接。

resolver 模块：解析Scrapy 的start\_urls属性下各个链接下的具体子链接， 并规范化

4：middlewares/，主要包括downloader中间件和spider中间件

downloader：包括用户代理，ip代理等中间件

spider：暂时未用

5：pipelines/，具体解析相应的抓取字段， 包括形成文件， 上传S3, 插入mongo等功能。对具体的新闻字段的解析， 使用了itemsresolver中的各个模块。

6：itemsresolver/，提供解析新闻各字段的模块：

titleresolver: 解析新闻标题字段的模块，主要TitleResolver类

dateresolver: 解析新闻发布时间字段模块，主要DateResolver类

authresolver: 解析新闻来源字段模块， 主要AuthResolver模块

textresolver: 解析新闻内容字段的模块， 主要TextResolver类

7：extractors/,主要是对scrapy的一个拓展Selector类和获取新闻字段的一些提取。

extensions模块：可以用来删除dom节点，主要是SlrExtension类。

extractor模块：主要提取新闻各个字段的类， NewsExtractor类

8：conf/, 各类新闻网站的配置文件

9：contrib/，一些公用的packages：

aws：利用boto模块上传文件到S3上。

Pyredis：redis队列， 将消息转换成文件，向集合里添加消息

Cached模块：主要将url添加到redis集合里

Convert 模块：将新加坡亚马逊的新闻推入redis队列， 并在北京亚马逊环境将新闻取出来， 生成文件。

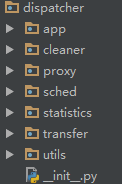
transfer 模块：将生成的文件推入NLP服务器

10：settings/, 关于新闻项目的配置和scrapy的配置

11: chema/, 基于实现分布式抓取及过滤功能

**二、关于job任务和调度模块**

1: job任务和调度的目录()dispatcher如下：



2：app/, run\_jobs所有任务启动的模块:

3: cleaner/, 定时清除redis过多的键值

4：proxy/，定时抓取IP代理， 并将其推入redis队列

5：sched/, 计算每个网站个时间段的新闻间隔时间并通过scrapyd调度该新闻项目：

base: 计算每个网站个时间段的新闻间隔时间

sched：根据计算的间隔时间分开调度各个网站

6：transfer/, 从队列接收消息产生文件，推送文件

receiver: 接收新加坡亚马逊

sender: 推送新闻文件到NLP服务器

7：utils/, 调度所需的工具

三、日志

目前日志在news\_log

四：优质新闻源的优先级，从以下几方面的因素参考：

1：手机端的展示良好（移动端新闻站点优先， 广告少）

2：新闻版权优先级（新闻是这家网站自己的， 而不是转载的）

3：数量多的

五：新闻分类：

热点新闻

公司新闻

宏观新闻

港股新闻

基金新闻

股评新闻

美股个股

美股宏观

美股股市

美股行业

京东新闻

京东美股

新三板

科技新闻

特别注意要点：

1： url解析

说明：首先从网站拿到初步的url, 由于这些urls可能不规范， 所以必须统一已处理已形成”http……..”

Package: news\_spiders/news\_spiders/urlsresolver

2：时间解析

说明：由于各网站网页时间的写法不一样， 必须将时间转换成14位字符串：“2160914100525”

Package: news\_spiders/news\_spiders/itemsresolver/dateresolver.py

3：新闻来源解析

说明：新闻来源有时与发布时间在同个标签， 有时在不同标签内， 不同标签处理起来会相对好一点， 如果在同一标签， 属于”时间 新闻来源”， 还是”新闻来源 时间”， 只需要改动pyq\_date\_author还是pyq\_author\_date。具体参考代码的解决方案

Package: news\_spiders/news\_spiders/itemsresolver/authresolver.py

4： 文本解析

说明：新闻文本解析， 每段新闻需要用“#&#”来区分；有时文本中含有一些冗余文本（比如：腾讯财经新闻的很多新闻末尾会来一段”微信扫一扫”, “扫描二维码”, 那么我们只需要保留该字样前面的文本即可），具体的实现逻辑可参照代码。

Package: news\_spiders/news\_spiders/itemsresolver/textresolver.py

5：过滤

说明：这里包括关键字（可见配置文件）过滤和url过滤(即相同的url在下一次不在抓取)

Package: news\_spiders/news\_spiders/utils/filter.py(关键字过滤)和 news\_spiders/news\_spiders/schema/dupefilter.py(包括了redis url 过滤)

6：统一环境文件传送

说明：这里有两种情况：

A: 北京亚马逊54.223.52.50（新闻抓取服务器）传到 54.223.46.84(文本分析服务器)， 将文件用相同目录从50传到84上

B: 新加坡亚马逊 54.251.56.190将分析后的文本推入redis队列里， 然后在北京亚马逊这边将文本信息解析出来，生成文件， 最终按照A的方式传到84上

注意：B不是最主要的

Package: 调度在news\_spiders/dispatcher/transfer/sender.py下，基本的文件传送在：news\_spiders\news\_spiders\contrib\transfer\smooth.py

7：部署问题(先scrapyd-deploy部署， 再上传工程启动项目)

说明： 一般有本机和北京亚马逊 54.223.52.50机器分别部署scrapyd服务， 然后在本机上通过

scrapyd-deply server\_amazon\_bj –p news\_spiders –v news\_spiders\_formal

将工程打包到 北京亚马逊 50机器(部署了scrapyd服务)， 然后上传工程在指定目录，启动 news\_spiders/dispatcher/app/run\_jobs.py脚本调度