**欢迎使用 Gather Platform 数据抓取平台**

[Readme in English](https://github.com/gsh199449/spider/tree/master/README-en.md)

欢迎加入 Gather Platform交流 QQ群 : 206264662

Gather Platform 数据抓取平台是一套基于[Webmagic](https://github.com/code4craft/webmagic)内核的,具有Web任务配置和任务管理界面的数据采集与搜索平台.具有以下功能

* 根据配置的模板进行数据采集
* 对采集的数据进行NLP处理,包括:抽取关键词,抽取摘要,抽取实体词
* 在不配置采集模板的情况下自动检测网页正文,自动抽取文章发布时间
* 动态字段抽取与静态字段植入
* 已抓取数据的管理,包括:搜索,增删改查,按照新的数据模板重新抽取数据

**Windows/Mac/Linux 全平台支持**

本系统需要如下依赖:

* JDK 8 及以上
* Tomcat 8.3 及以上
* Elasticsearch 5.0

**部署方式**

本系统提供一份预编译版本和配置好的依赖环境,只需从百度云下载,按照步骤安装即可使用.

**1. 使用预编译版本抓取数据**

* 从百度云[下载](https://pan.baidu.com/s/1i4IoEhB) 密码: v3jm, 预编译安装包和依赖环境,\*nix用户下载 elasticsearch-5.0.0.zip ,windows用户请下载 elasticsearch-5.0.0-win.zip 包

Windows用户**注意**:如果在启动elasticsearch时发生elasticsearch.bat闪退的问题请修改 JAVA\_HOME/jre/lib/security/java.policy 文件,在 grant 段落里面加入

permission java.io.FilePermission "ES安装路径/\*", "read";

除此之外,ES和tomcat请勿安装到含有空格或者中文等字符的路径下面.

* 安装JDK 8 ,从[ORACLE](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)下载JDK
* 解压elasticsearch5.0.0.zip
* 进入bin文件夹,若是\*nix环境运行 elasticsearch ,windows环境运行 elasticsearch.bat
* 使用浏览器访问 http://localhost:9200,显示如下内容则证明elasticsearch安装成功
* {
* "name" : "AQYRo1f",
* "cluster\_name" : "elasticsearch",
* "cluster\_uuid" : "0LJm-YogQ2qgLLznrlvWwQ",
* "version" : {
* "number" : "5.0.0",
* "build\_hash" : "080bb47",
* "build\_date" : "2016-11-11T22:08:49.812Z",
* "build\_snapshot" : false,
* "lucene\_version" : "6.2.1"
* },
* "tagline" : "You Know, for Search"

}

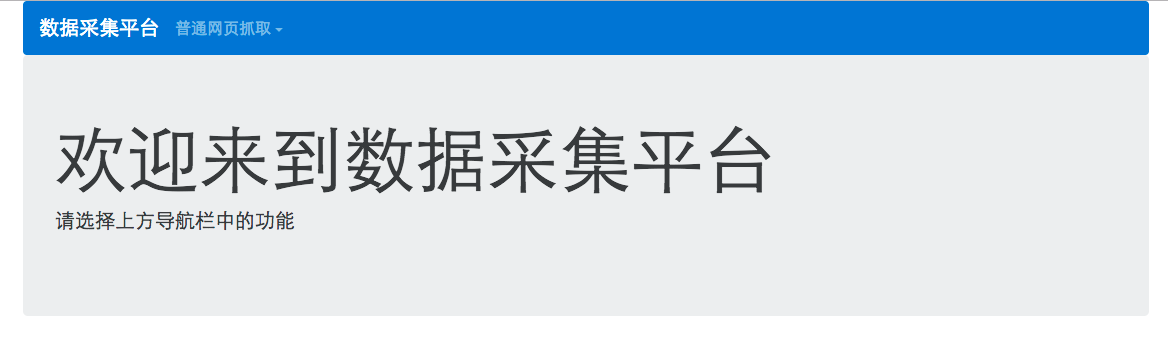
* 解压apache-tomcat-8.zip,将spider.war放入Tomcat下面的webapp文件夹
* 进入tomcat目录下的bin文件夹,若是\*nix环境运行 startup.sh ,windows环境运行 startup.bat
* 然后使用浏览器访问 http://localhost:8080/spider 打开数据采集平台Web控制台

**2. 手工编译安装**

* 安装 JDK 8 以上版本, [ORACLE](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)
* 下载并安装Elasticsearch 5.0, [elastic.co](https://www.elastic.co/downloads/past-releases/elasticsearch-5-0-0)
* 安装ansj-elasticsearch插件, [github](https://github.com/NLPchina/elasticsearch-analysis-ansj)
* 运行Elasticsearch
* 安装Tomcat 8, [Apache Tomcat](http://tomcat.apache.org/)
* 下载本项目源码包
* 执行 mvn package 编译打包
* 将spider.war放入Tomcat下面的webapp文件夹
* 运行tomcat

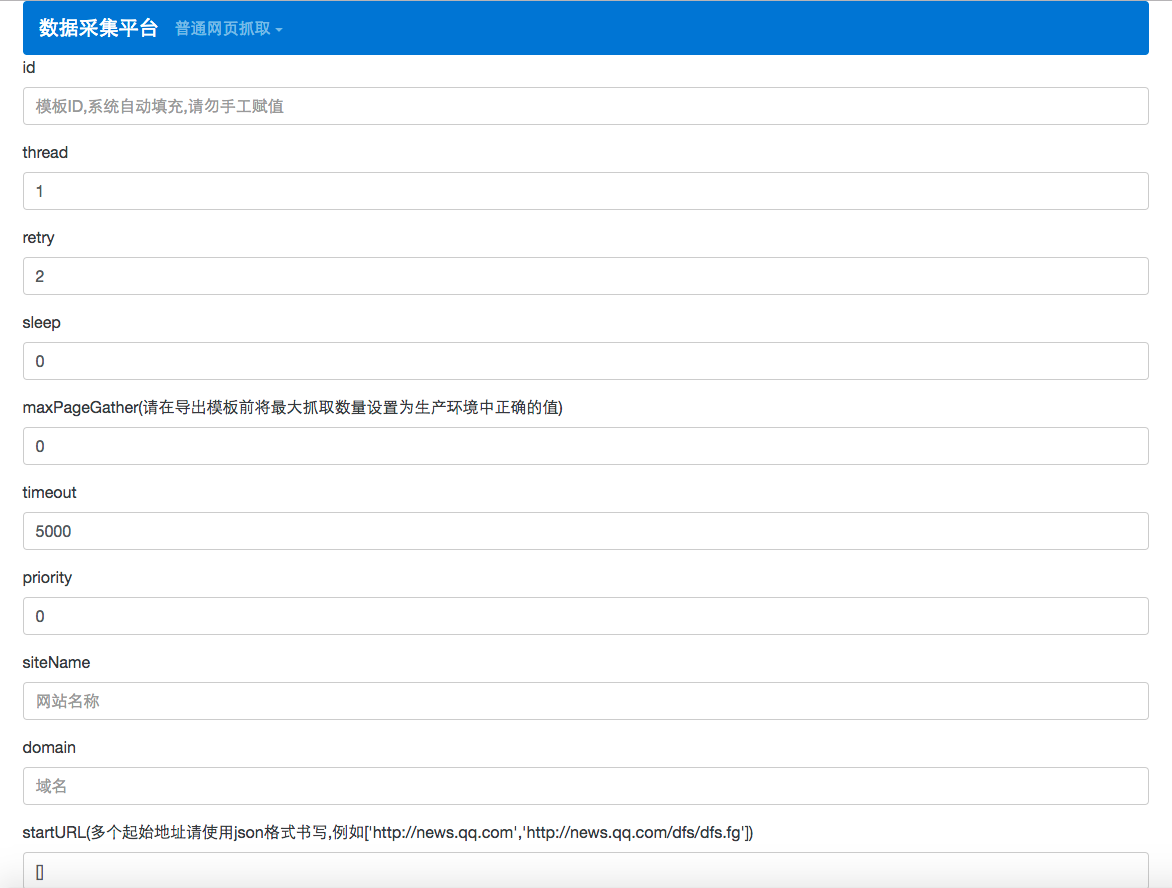
**使用方法**

部署完成后打开浏览器,访问 http://localhost:8080/spider 打开采集平台首页,点击导航栏的下拉菜单选择功能.

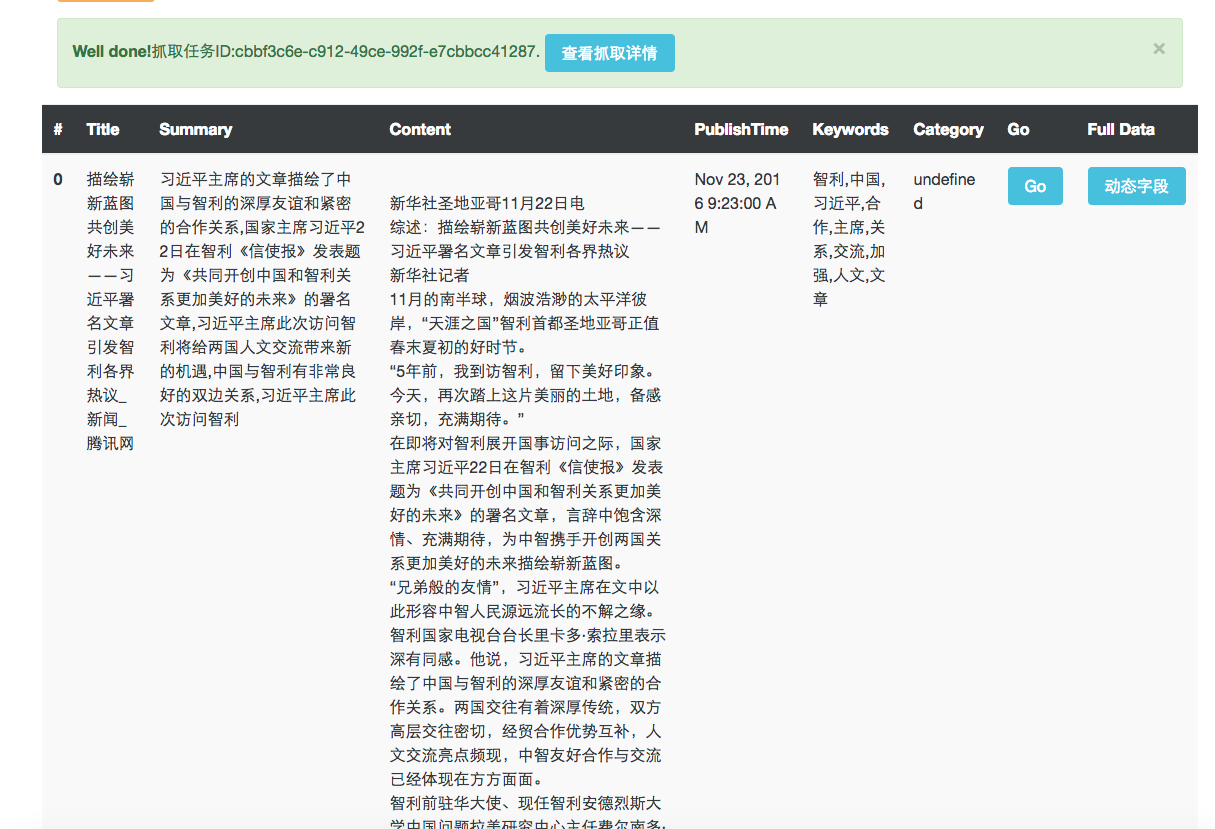
[](https://raw.githubusercontent.com/gsh199449/spider/master/doc/imgs/home.png)

**配置爬虫模板**

在导航栏的下拉菜单中点击 编辑模板 按钮,在这个页面中可完成一个爬虫的所有配置,具体每一个配置项的说明见每一个输入框的提示.

[](https://raw.githubusercontent.com/gsh199449/spider/master/doc/imgs/spiderinfo.png)

爬虫模板配置完成后,点击下面的 采集样例数据 按钮,稍等片刻即可在下方展示根据刚刚配置的模板抓取的数据,如果数据有误在上面的模板中进行修改,然后再次点击 采集样例数据 按钮即可重新抓取.

[](https://raw.githubusercontent.com/gsh199449/spider/master/doc/imgs/testSpiderinfo.png)

注意,在对于爬虫模板没有完全的把握之前请勿选择爬虫模板下方的几个 是否网页必须有XXX 的配置项.以文章的标题为例,因为如果文章标题的配置项(即为title)配置有误,爬虫就无法抓取到网页的标题,如果这时再选中了 是否网页必须有标题 的话,就会导致爬虫无限制的进行抓取.

[](https://raw.githubusercontent.com/gsh199449/spider/master/doc/imgs/need.png)

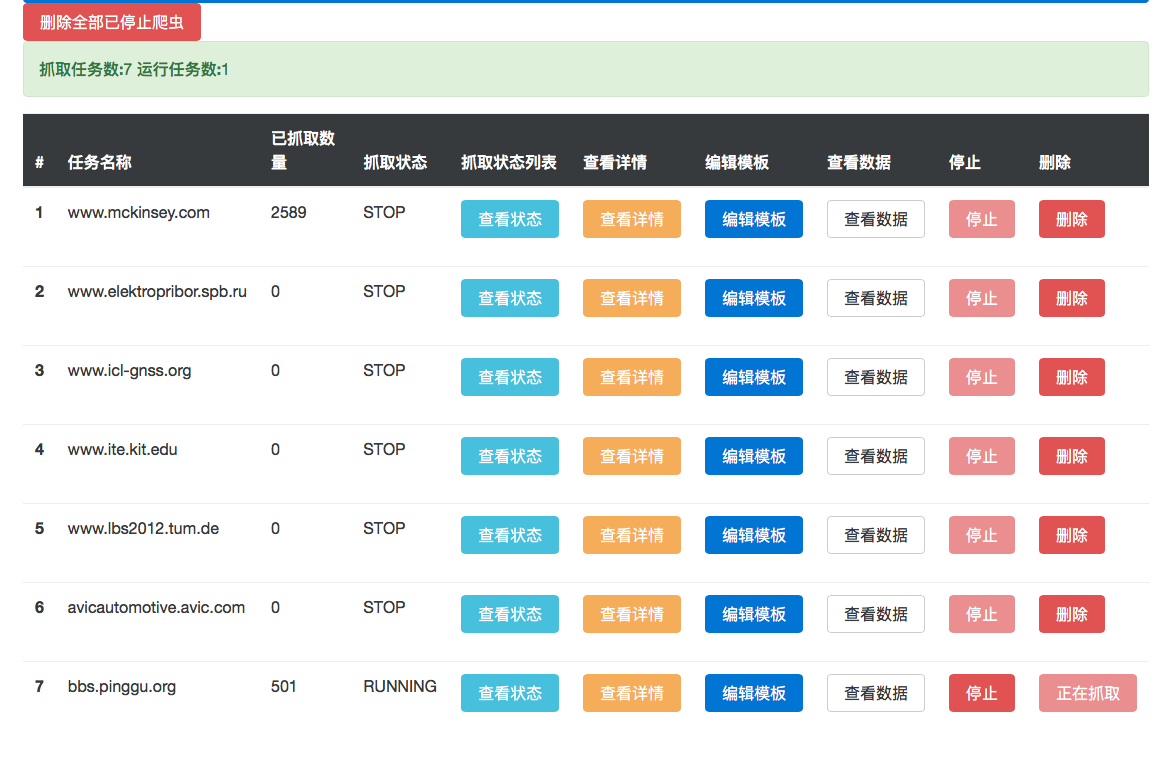
当模板配置完毕,即可点击下方的 导出模板 按钮,这时下方的大输入框中显示的Json格式的文字即为爬虫模板,可以将段文字保存到文本文件中,以便以后使用,也可以点击 存储此模板 对这个模板进行存储,以后可在本平台的 爬虫模板管理系统 中查找.

**快速上手**

本平台在examples文件夹中给出了两个抓取腾讯新闻的示例,这两个一个是使用预定义的发布时间抓取规则,另外一个是使用系统自动探测文章的发布时间. 以预定义的爬虫模板为例,打开[news.qq.com.json](https://github.com/gsh199449/spider/tree/master/examples/news.qq.com.json),将文件内容全部拷贝至爬虫模板编辑页面最下方的大输入框中,点击自动填充.这时爬虫配置文件中的爬虫模板信息就被自动填充进上面的表格了.然后点击抓取样例数据按钮,稍等片刻即可在当前页面下方看到通过这个模板抓取的新闻数据了. 如果模板配置的有问题,导致长时间卡在获取数据页面,请转至爬虫监控页面,将刚刚提交的这个抓取任务停止即可.

**爬虫监控**

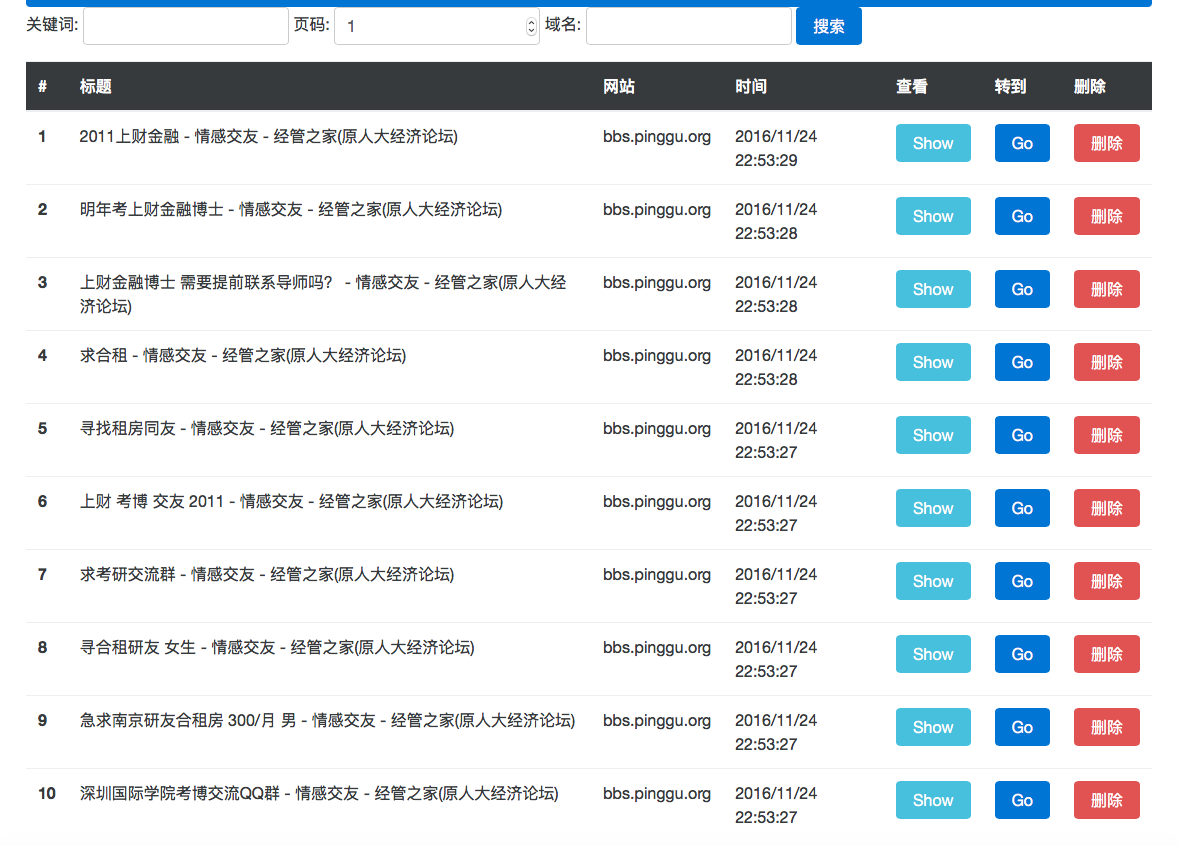
在导航栏中点击查看进度就可以看到当前爬虫的运行状况,在这个界面中可以实现对爬虫的停止,删除,查看进度,查看已抓取的数据,查看模板等操作.

[](https://raw.githubusercontent.com/gsh199449/spider/master/doc/imgs/spiderList.png)

注意,按照采集平台默认配置,这里的所有爬虫运行记录将在每两个小时对于已经完成的爬虫进行删除.如果不想让系统定时自动删除任何爬虫记录,或者改变删除记录的时间周期,请参阅高级配置中对于配置文件的解释部分.

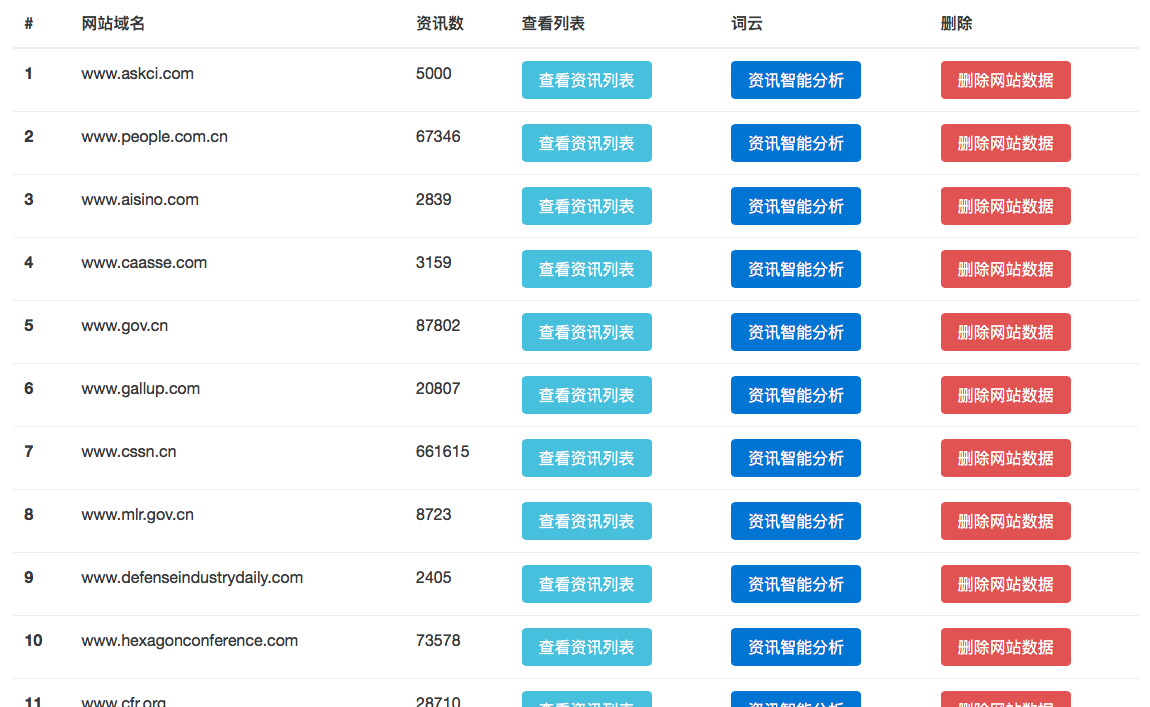
**数据管理与搜索**

点击导航栏下拉菜单中的搜索,即可看到目前Elasticsearch库中存储的所有网页数据,这些网页数据在默认情况下是按照抓取时间进行排序的,也就是说,在导航栏点击 搜索 之后,展示的第一条数据就是最新抓取的数据.

[](https://raw.githubusercontent.com/gsh199449/spider/master/doc/imgs/search.png)

在搜索页面是上方可以通过输入关键词对所有已抓取的网页数据进行搜索,也可以指定网站域名查看指定网站的所有数据.如果是指定关键词进行搜索则搜索结果是按照与输入的关键词的相关度进行排序的,如果是输入域名,查看某一网站的所有数据则按照抓取时间进行排序,最新抓取的数据在最上方.

点击导航栏中的 网站列表 按钮即可查看目前已经抓取的数据中都是那些网站的信息,对于每个网站都可以点击 查看数据列表 按钮查看该网站的所有数据.点击 删除网站数据 即可删除该网站下的所有数据.

[](https://raw.githubusercontent.com/gsh199449/spider/master/doc/imgs/domainList.png)

**高级使用**

**动态字段和静态字段**

在配置网页模板时,有一个 添加动态字段 按钮,这个功能是为了抓取那些不在预设字段里的其他字段所设计.举例来说,目前爬虫模板可以配置的预设字段有:标题,正文,发布时间等等.如果我们想抓取文章的作者或者文章的发布文号,这时就需要使用动态字段来实现.

[](https://raw.githubusercontent.com/gsh199449/spider/master/doc/imgs/dynamic.png)

点击 添加动态字段 按钮,在弹出的输入框中输入要抓取的字段名称,我们以要抓取文章的作者为例,在框中输入author. 注意这个动态字段的名称必须使用英文名称.之后再模板编辑页面就多出来的两个输入框,一个是author Reg,一个是author XPath,其中一个是配置作者字段的正则表达式,另一个是配置作者字段的XPath表达式,这两个选其一即可.

静态字段使用方法与动态字段类似,但是与动态字段不同的是,静态字段相对于爬虫模板来说是静态的.也就是说这个值在配置模板的阶段就是预设好的,通过这个模板抓取的所有数据里面都会带有这个字段和预设的这个值.这个功能主要是方便二次开发人员在数据存储于搜索时的使用.

**使用Lucene Query进行数据查询**

在数据查询页面进行数据查询时,在关键词输入框中输入的检索词默认是在文章正文中进行检索.如果在这个框中输入 title: 中国 的含义是在所以文章的标题中检索带有中国的网页.支持的字段名称有(括号前为字段名称,括号内为字段的含义):

* content(正文)
* title(标题)
* url(网页链接)
* domain(网页域名)
* spiderUUID(爬虫id)
* keywords(文章关键词)
* summary(文章摘要)
* publishTime(文章发布时间)
* category(文章类别)
* dynamic\_fields(动态字段)

**同一网站不同模板的情况**

针对同一网站可以有不同的抽取模板的问题,可以通过配置另外的模板进行解决.

**高级配置**

项目的配置文件在spider.war/WEB-INF文件夹下

**输出网页数据至Redis Channel**

找到 mvc-dispatcher-servlet.xml 配置文件,增加redis数据输出管道:

<property name="pipelineList">

<list>

<!--Redis输出-->

<ref bean="commonWebpageRedisPipeline"/>

<ref bean="commonWebpagePipeline"/>

</list>

</property>

在 staticvalue.json 配置文件中,修改如下内容:

"needRedis": false,

"redisPort": 6379,

"redisHost": "localhost",

"webpageRedisPublishChannelName": "webpage",

* needRedis设置为true将在系统启动时检查redis配置
* redisPort redis的端口
* redisHost redis的服务器地址
* webpageRedisPublishChannelName redis发布通道的名称

**输出网页数据至任意数据源**

写一个类实现 Pipeline 接口,然后在 mvc-dispatcher-servlet.xml 配置文件中配置这个数据处理类, 爬虫框架会把每一个采集并处理好的的网页传入 process 方法,然后通过自己的代码将这些网页数据存储至你想要的位置即可,可以参考本平台实现的[Redis pipeline](https://github.com/gsh199449/spider/blob/master/src/main/java/com/gs/spider/dao/CommonWebpageRedisPipeline.java).本平台默认的ES输出[CommonWebpagePipeline](https://github.com/gsh199449/spider/blob/master/src/main/java/com/gs/spider/dao/CommonWebpagePipeline.java)中有一个 convertResultItems2Webpage 便利方法,可以将Webmagic框架的 ResultItems 对象转换为一个 Webpage 对象方便处理.

**配置文件解释**

本系统配置文件名称为 staticvalue.json :

{

"esHost": "localhost",

"esClusterName": "elasticsearch",

"commonsIndex": "commons",

"maxHttpDownloadLength": 1048576,

"commonsSpiderDebug": false,

"taskDeleteDelay": 1,

"taskDeletePeriod": 2,

"limitOfCommonWebpageDownloadQueue": 100000,

"needRedis": false,

"redisPort": 6379,

"redisHost": "localhost",

"webpageRedisPublishChannelName": "webpage",

"commonsWebpageCrawlRatio": 2

}

* esHost:es服务器地址
* esClusterName:es集群名称
* commonsIndex:网页数据在es中存储的index名称
* maxHttpDownloadLength: 如果网页超过这个大小则不再下载这个网页
* commonsSpiderDebug:置为true则在爬虫管理系统的爬虫运行日志中可以看到更多的错误信息
* taskDeleteDelay: 自动删除任务记录延时几小时启动
* taskDeletePeriod: 每几小时删除一次已完成的任务记录
* limitOfCommonWebpageDownloadQueue: 最大网页下载队列长度
* commonsWebpageCrawlRatio: 如果抓取的网页超过需要网页commonsWebpageCrawlRatio倍,爬虫退出