# Uyint爬虫配置文档

1. **操作流程说明**

Uyint的初始界面包含四个模块：分别是任务模块、监控模块、配置模板模块和流程模板模块。

图1 模块结构图

### 1.1任务模块



图1.1.1 任务模块展示图

任务模板显示的是已配置完成的任务信息，展示的信息包括任务名称、爬取间隔时间、周期表达式、间隔类型、爬虫类型、优先级、配置是否完成这几个属性。点击右上角的新建任务能够新建爬虫任务。任务模板中操作项中支持的操作包括如下三个：

 启动任务爬虫或停止任务爬虫

 修改当前任务的基本信息，如任务名称、间隔时间、间隔类型、cookie、爬虫类型等

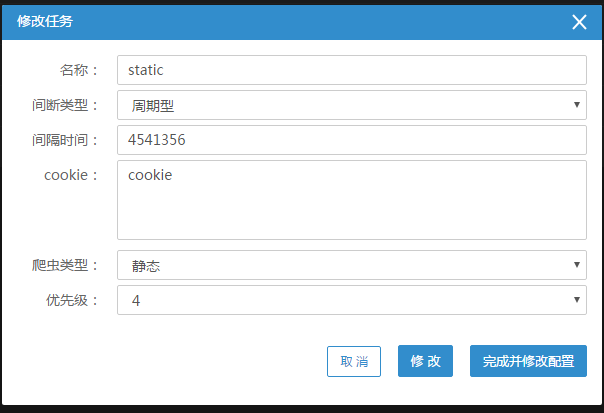


图1.1.2 修改任务基本信息

* 任务名称：配置任务的名称
* 间隔类型：周期型（用户设置的间隔时间t 爬虫每隔t时间启动爬取）、定时型（按照用户配置的周期表达式来运行采集任务）
* 间隔时间（周期型）：爬虫任务间隔时间
* 周期表达式（定时型）：具体配置参见附录1
* 爬虫类型：静态爬虫（爬取静态网页内容，静态网页指页面上的内容能够在html源码中得到）、动态爬虫（动态爬虫基于浏览器进行数据采集，动态网页指通过js代码动态展示内容，有些内容无法通过源码获得）

 修改当前任务的详细配置信息 点击此按钮会进入详情配置页面（下图）

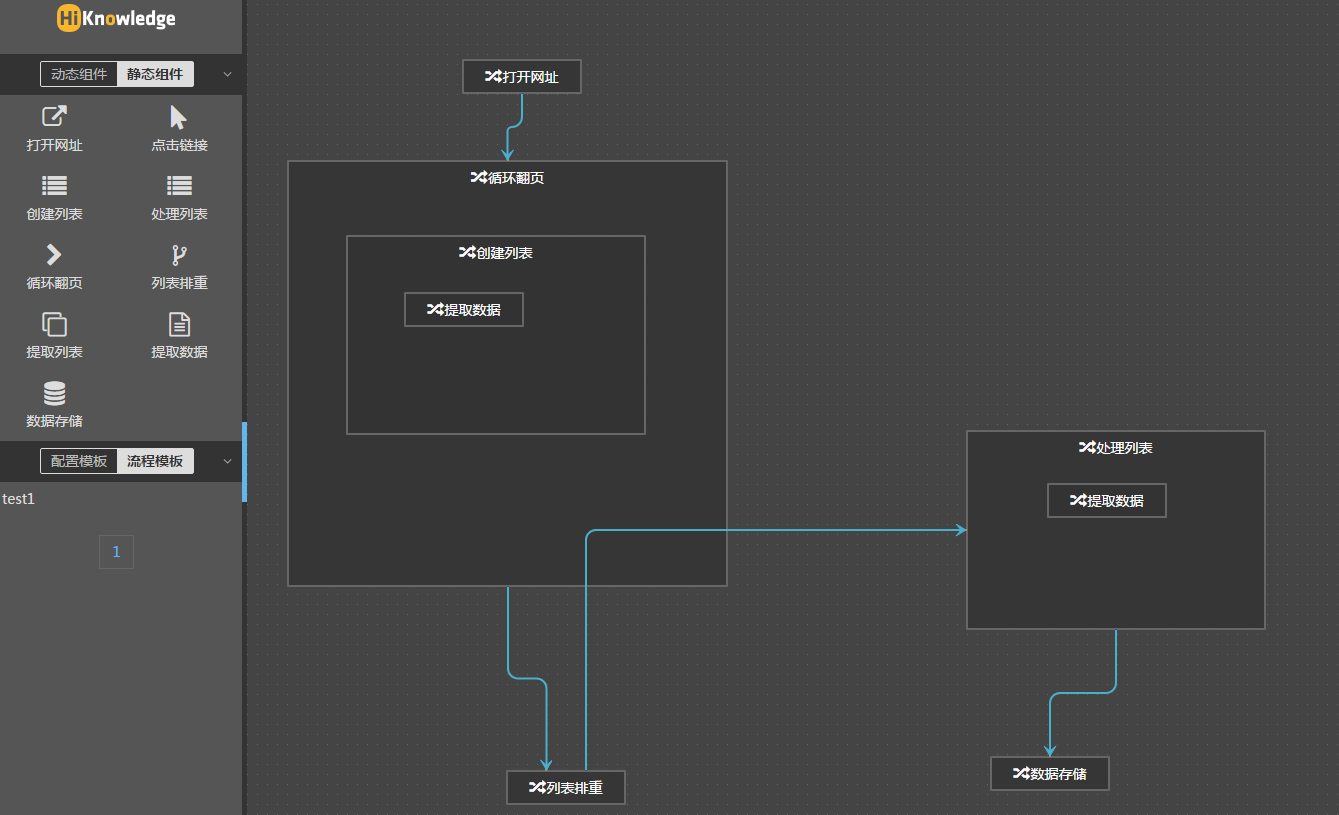


图1.1.3 详情配置图

### 1.2 监控模块

监控模块展示的是爬虫信息的爬取情况，展示的信息包括爬虫ID、爬虫名称、爬取大小、汇报时间。

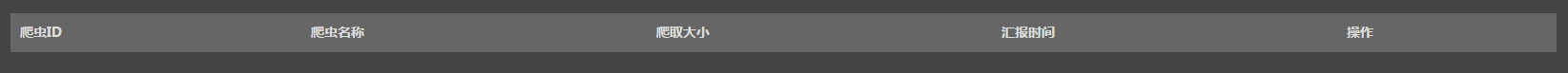


图1.2.1 监控信息展示图

爬虫ID：系统自动生成的标识

爬虫名称：该采集任务的名称

爬取大小： 当天该任务爬取的总数据量 单位是条数

汇报时间：任务最后一次爬取时间

监控模块中，对于爬虫任务支持基于日期的检索方式。

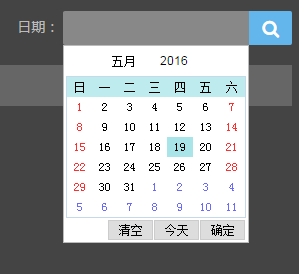


图1.2.2 搜索展示图

### 1.3 配置模板模块

当用户在爬取结构相类似的页面时，可以选择在配置模板模块中生成配置文件，以便在后续的任务配置界面更加方便地引用，避免重复输入相同的信息。配置模块展示的信息包括模板名称和模板支持的爬虫类型（**注意：模板只支持相对应的爬虫任务配置，例如静态爬虫只支持静态模板。如果更改了爬虫的类型，之前的模板配置信息也会清空**）



图1.3.1 配置模块展示图

点击右上角的新建模板可以新建模板，填入基本信息就可

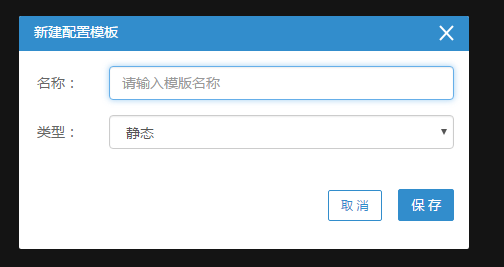


图1.3.2 新建配置模板

配置模板支持的操作如下：

 删除模板

修改基本信息 名称 爬虫类型

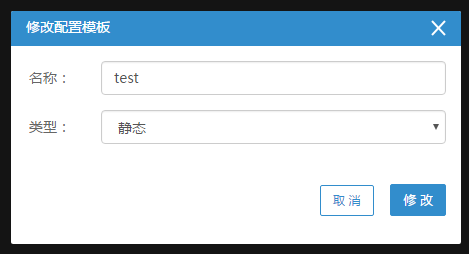


图1.3.3 修改配置模板

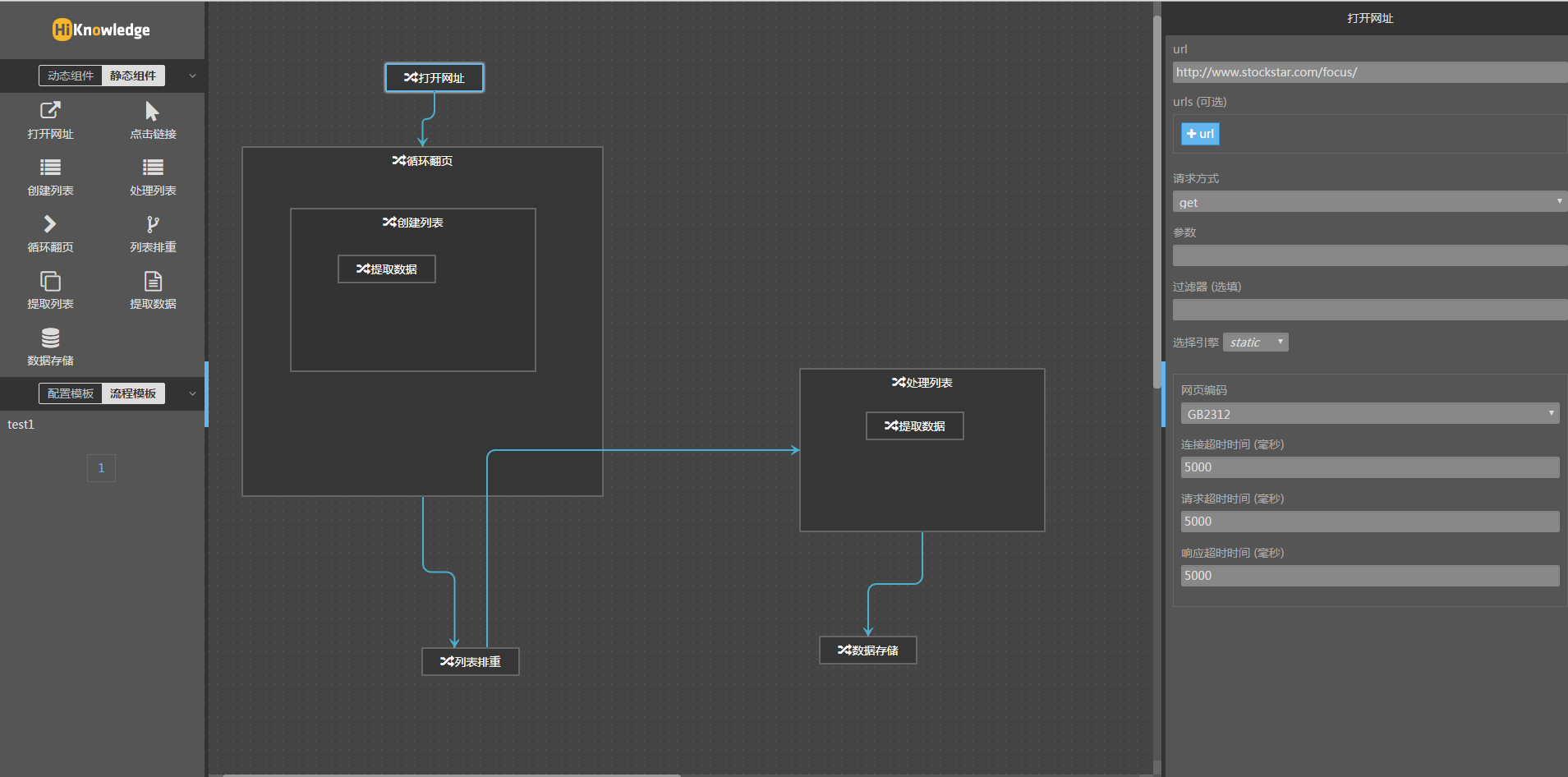
 进入模板详情配置页面

### 1.4 流程模板模块

流程模板模块与配置模板块操作方法相同。流程模板主要存储一些固化的爬取流程，如先打开页面，然后爬取列表，最后爬取列表内容这类固化的流程。

### 1.5 详细配置页面介绍

详细配置页面分为三栏展示：

图1.5.1详细配置页面

第一栏：组件展示和模板选择



图1.5.2 组件栏

用户根据自己的需求拖动相应的组件进入主配置页面，同时也可以选择下方的配置模板和流程模板快速生成配置文件。

第二栏：主配置栏

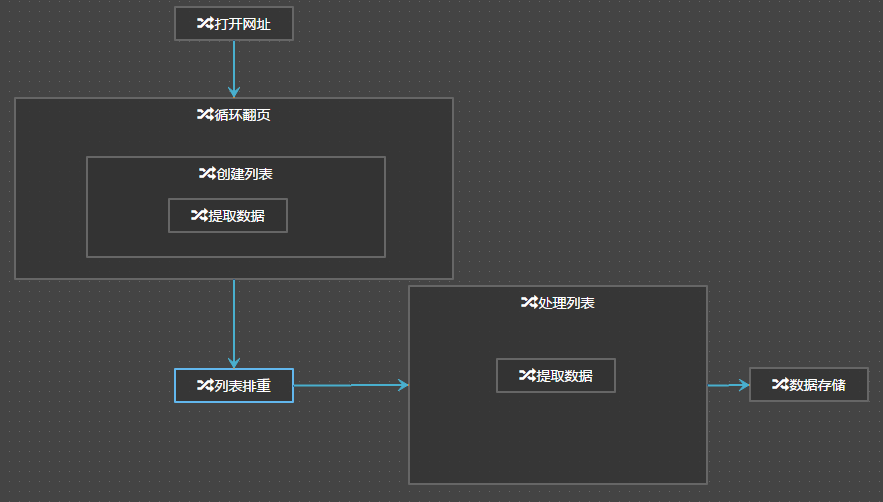


图1.5.3 主配置栏

主配置栏完成整个配置文件的可视化生成，用户通过连线来决定组件之间执行的先后顺序。

第三栏：组件属性栏

点击相应的组件，组件属性栏就显示该组件的详细属性信息。

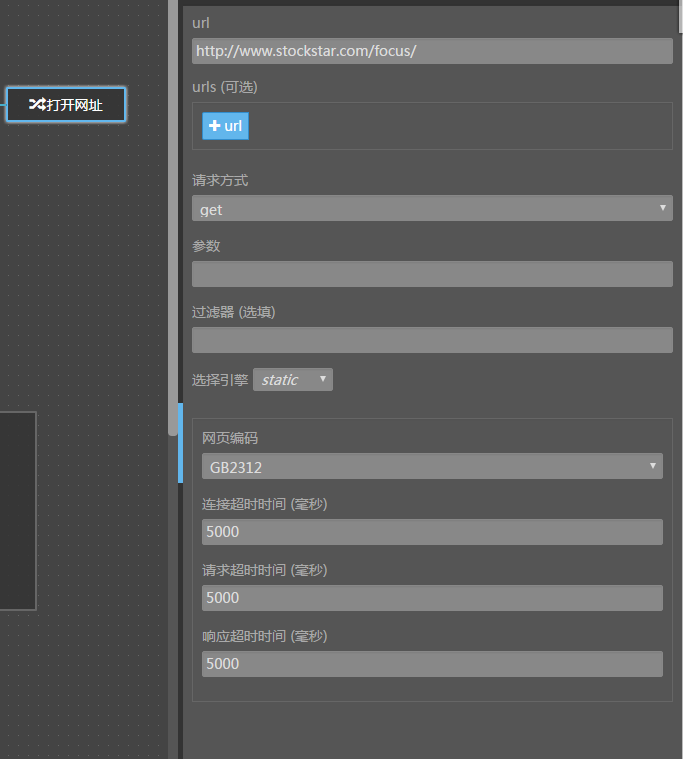


图1.5.4 组件属性栏

1. **详细配置属性说明**

### 静态爬虫组件描述

1. 打开（不可嵌套）

打开网址能够获得要爬取页面的整个html源码。

|  |  |  |
| --- | --- | --- |
| url | | url地址 可用$字符引用之前的提取字段 例如$url 即引用之前的url字段值 |
| Urls | | 多个url地址 |
| 请求方式 | | Get/Post |
| 参数（选填） | | post的请求参数 以分号分割的键值对 键值之间用等号 |
| 过滤器（选填） | | 配一个正则表达式，第一个捕获组就是要保留的数据 |
| 引擎选择（静态） |  | |
|  | 网页编码 | 网页的编码选择（可选utf-8，gb2312，gbk，gb18030） |
| 连接超时时间（connection timeout） | 默认5000毫秒 |
| 请求超时时间（request timeout） | 默认5000毫秒 |
| 响应超时时间（socket timeout） | 默认5000毫秒 |
| 引擎选择（自定义） |  | |
|  | 调用类名 | 自定义方法的class名称 |

1. 循环翻页（可嵌套提取数据、点击链接、排重、创建列表）

循环翻页要爬取页面。通过打开网址后的页面获得页面的信息，寻找到列表页url信息的规律或者定位到页面上下一页地址的url来获得循环翻页的效果。

|  |  |  |
| --- | --- | --- |
| 引擎选择（静态） |  | |
|  | 网页编码 | 网页的编码选择（可选utf-8，gb2312，gbk，gb18030） |
| 连接超时时间（connection timeout） | 默认5000毫秒 |
| 请求超时时间（request timeout） | 默认5000毫秒 |
| 响应超时时间（socket timeout） | 默认5000毫秒 |
| 引擎选择（自定义） |  | |
|  | 调用类名 | 自定义方法的class名称 |
| 规则翻页 |  | |
|  | 翻页规则 | 用currentPage变量计算下一页的页码数 currentPage从1开始，每次增加1。计算值替换翻页url中的（pn） |
| 翻页url | （pn）作为占位符 |
| 下一页翻页 |  | |
|  | 方式 | 定位下一页元素的方式（Xpath，Regex，CSS，Common） |
| 选择器 | 定位元素的路径（元素路径相对于上一层嵌套组件获得的元素路径 例如上一层已经定位到div标签 想要提取div中的a 只需在此填写a 而不用填写div > a） |
| 前缀检查（可选） | 判断下一页url是否需要添加前缀 填入检查的前缀字符串 |
| 前缀（可选） | 依赖于前缀检查 如需要添加前缀 对于下一页的url添加前缀 |
| 请求类型 | | Get/post |
| 参数 | | post的请求参数 以分号分割的键值对 键值之间用等号 |
| 过滤器（选填） | | 配一个正则表达式，第一个捕获组就是要保留的数据 |
| 最大页数 | | 翻页的最大页数 |
| 翻页间隔时间 | | 每次发送翻页请求时的间隔时间 |
| 循环（自定义） | | 自定义方法的类名 |

1. 点击链接（不可嵌套）

点击一个元素的链接，获取链接的源码信息。

|  |  |  |
| --- | --- | --- |
| 引擎选择（静态） |  | |
|  | 网页编码 | 网页的编码选择（可选utf-8，gb2312，gbk，gb18030） |
| 连接超时时间（connection timeout） | 默认5000毫秒 |
| 请求超时时间（request timeout） | 默认5000毫秒 |
| 响应超时时间（socket timeout） | 默认5000毫秒 |
| 引擎选择（自定义） |  | |
|  | 调用类名 | 自定义方法的class名称 |
| linkDom |  | |
|  | 方式 | 定位链接元素的方式（Xpath，Regex，CSS，Common） |
| 选择器 | 定位链接元素的路径（元素路径相对于上一层嵌套组件获得的元素路径 例如上一层已经定位到div标签 想要提取div中的a 只需在此填写a 而不用填写div > a） |
| 前缀（可选） | 对于定位到的元素添加前缀 |
| linkUrl |  | |
|  | url | 进行搜索的url |
| 占位符（可选） | 搜索url中的关键词占位符，占位符自定义 |
| 替换占位符（可选） | 替换占位符的字符串--关键词 |
| referField |  | |
|  | 字段 | 点击的链接url对应已提取的字段 |
| 请求类型 | | Get/post |
| 参数 | | post的请求参数 以分号分割的键值对 键值之间用等号 |
| 过滤器 | | 配一个正则表达式，第一个捕获组就是要保留的数据 |
| 最大页数 | | 翻页的最大页数 |
| 翻页间隔时间 | | 每次发送翻页请求时的间隔时间 |
| 点击（自定义） | | 自定义方法的类名 |

1. 创建列表（可嵌套提取数据、点击链接、排重、创建列表）

定位每一页的列表元素。通常依赖于循环翻页提取每一页上的列表元素信息。

|  |  |
| --- | --- |
| 方式 | 定位列表元素的方式（Xpath，Regex，CSS，Common） |
| 选择器 | 定位列表元素的路径（元素路径相对于上一层嵌套组件获得的元素路径 例如上一层已经定位到div标签 想要提取div中的a 只需在此填写a 而不用填写div > a） |
| 创建列表（自定义） | 自定义方法的类名 |

1. 处理列表（可嵌套提取数据、点击链接、排重、创建列表）

处理每一个列表url。对于每一个之前提取到的列表url进行进一步处理。

|  |  |
| --- | --- |
| 元素处理间隔时间（毫秒） | 每个列表元素处理的间隔时间，单位毫秒 |
| 处理列表方式（自定义） | 自定义方法的类名 |

1. 列表排重（不可嵌套）

对于要爬取的数据进行去重。可根据url，标题等之前配置的属性进行去重。

|  |  |  |
| --- | --- | --- |
| 引擎选择（mongodb） |  | |
|  | 主机 | 主机地址 默认localhost |
| 端口 | 端口号 默认27017 |
| 用户（可选） | 用户 |
| 密码（可选） | 密码 |
| 数据库名 | Mongo的数据库名称 |
| 集合名 | Mongo的collection名称 |
| 引擎选择（jdbc） |  | |
|  | 驱动 | Jdbc的驱动方式 |
| url | 数据库的地址 |
| 用户 | 用户名 |
| 密码 | 密码 |
| Table | 表名 |
| 引擎选择（redis） |  | |
|  | 主机 | Redis主机地址 |
| 端口 | 端口号 |
| 数据库索引编号 | 默认0 |
| 键 | 排重set的键名 |
| 点击（自定义） | | 自定义方法的类名 |
| 排重方式 | | MD5 |
| 属性字段名 | | 要排重的字段名 |
| 数据库映射字段名 | | 数据库中相对应的字段名 |

1. 提取列表（可嵌套点击链接、创建列表、循环翻页）

提取网页中的列表元素

|  |  |
| --- | --- |
| 字段 | 字段名 |
| 提取列表（自定义方法） | 自定义方法的类名 |

1. 提取数据（不可嵌套）

可提取的信息包括该列表url页面上html所包含的全部信息。并可提取多个字段信息。

|  |  |  |
| --- | --- | --- |
| 标题（可选） | | 新闻标题 |
| 是否提取源码 | | 默认false |
| 过滤器 | | 配一个正则表达式，第一个捕获组就是要保留的数据 |
| 抽取字段 |  | |
|  | 字段名 | 抽取字段的名称 |
| 方式 | 定位字段元素的方式（Xpath，Regex，CSS，Common，Constant，Index） |
| 选择器 | 定位元素的路径（元素路径相对于上一层嵌套组件获得的元素路径 例如上一层已经定位到div标签 想要提取div中的a 只需在此填写a 而不用填写div > a） |
| 属性 | 元素中的属性 |
| 前缀检查（可选） | 判断抽取字段是否需要添加前缀 填入检查的前缀字符串 |
| 前缀（可选） | 依赖于前缀检查 如需要添加前缀 对于抽取字段添加前缀 |
| 后缀检查（可选） | 判断抽取字段是否需要添加后缀 填入检查的后缀字符串 |
| 后缀（可选） | 依赖于后缀检查 如需要添加后缀 对于抽取字段添加后缀 |
| 字符串截取（可选） | 截取表达式 以,分割 |
| 组号（可选） | 通过正则表达式捕获的组号 |
| 常量值（可选） | 常量 |
| 节点位置（可选） | 通过index方法获得的节点位置 从0开始 |
| 格式器（可选） | 可选time trim longtime |
| 提取数据（自定义） | | 自定义方法的class名称 |

1. 数据存储（不可嵌套）

将爬取的字段数据存入数据库。

|  |  |  |
| --- | --- | --- |
| 引擎选择（mongodb） |  | |
|  | 主机 | 主机地址 默认localhost |
| 端口 | 端口号 默认27017 |
| 用户（可选） | 用户 |
| 密码（可选） | 密码 |
| 数据库名 | Mongo的数据库名称 |
| 集合名 | Mongo的collection名称 |
| 引擎选择（jdbc） |  | |
|  | 驱动 | Jdbc的驱动方式 |
| url | 数据库的地址 |
| 用户 | 用户名 |
| 密码 | 密码 |
| Table | 表名 |
| 引擎选择（elasticsearch） |  | |
|  | 集群名 | Es集群名称 |
| 主机 | Es主机地址 |
| 端口 | Es端口号 |
| 索引名称 | 索引名称 |
| 类型名称 | 类型名称 |
| 字段名 | | 要存储的字段名 |
| 数据库映射字段名 | | 数据库中相对应的字段名 |

### 动态爬虫组件描述

1. 打开（不可嵌套）

通过浏览器打开动态加载的网页。

|  |  |  |
| --- | --- | --- |
| url | | url地址 可用$字符引用之前的提取字段 例如$url 即引用之前的url字段值 |
| Urls | | 多个url地址 |
| 请求方式 | | Get/Post |
| 参数 | | post的请求参数 |
| 过滤器（选填） | | 配一个正则表达式，第一个捕获组就是要保留的数据 |
| 引擎选择（静态） |  | |
|  | 网页编码 | 网页的编码选择（可选utf-8，gb2312，gbk，gb18030） |
| 连接超时时间（connection timeout） | 默认5000毫秒 |
| 请求超时时间（request timeout） | 默认5000毫秒 |
| 响应超时时间（socket timeout） | 默认5000毫秒 |
| 引擎选择（动态） |  | |
|  | 脚本超时时间（script timout） | 默认5000毫秒 |
| 加载超时时间（load timeout） | 默认5000毫秒 |
| 等待超时时间（explicit wait） | 默认5000毫秒 |

1. 点击链接（不可嵌套）

点击一个元素的链接，获取链接的源码信息。

|  |  |
| --- | --- |
| 方式 | 定位链接的方式（Id，Class，Css，TagName，Name，LinkText，ParitialLinkText，Xpath，JavaScript） |
| 选择器 | 定位元素的路径（元素路径相对于上一层嵌套组件获得的元素路径 例如上一层已经定位到div标签 想要提取div中的a 只需在此填写a 而不用填写div > a） |
| 脚本超时时间（script timout） | 默认5000毫秒 |
| 加载超时时间（load timeout） | 默认5000毫秒 |
| 等待超时时间（explicit wait） | 默认5000毫秒 |

1. 创建列表（可嵌套提取数据、点击链接、排重、创建列表）

定位每一页的列表元素。通常依赖于循环翻页提取每一页上的列表元素信息。

|  |  |
| --- | --- |
| 方式 | 定位列表元素的方式（Id，Class，Css，TagName，Name，LinkText，ParitialLinkText，Xpath，JavaScript） |
| 选择器 | 定位元素的路径（元素路径相对于上一层嵌套组件获得的元素路径 例如上一层已经定位到div标签 想要提取div中的a 只需在此填写a 而不用填写div > a） |

1. 处理列表（可嵌套打开、页面滚动、提取列表、提取数据、点击链接、排重、创建列表）

处理每一个列表url。对于每一个之前提取到的列表url进行进一步处理。

|  |  |
| --- | --- |
| 处理间隔时间 | 每个列表元素处理间隔的时间 单位毫秒 |

1. 循环翻页（可嵌套打开、页面滚动、提取列表、提取数据、点击链接、排重、创建列表）

循环翻页要爬取页面。通过打开网址后的页面获得页面的信息，寻找到列表页url信息的规律或者定位到页面上下一页地址的url来获得循环翻页的效果。

|  |  |  |
| --- | --- | --- |
| 最大页 | | 循环翻页的最大页数 |
| 翻页间隔时间 | | 每次请求下一页的间隔时间 |
| 当前页 | | 默认1 |
| 脚本超时时间（script timout） | | 默认5000毫秒 |
| 加载超时时间（load timeout） | | 默认5000毫秒 |
| 翻页策略 |  | |
|  | More | 点击更多方式 |
| Scroll | 滚动翻页 （方式中只支持JavaScript） |
| next | 下一页翻页 |
| 方式 | | 定位列表元素的方式（Id，Class，Css，TagName，Name，LinkText，ParitialLinkText，Xpath，JavaScript） |
| 选择器 | | 定位元素的路径（元素路径相对于上一层嵌套组件获得的元素路径 例如上一层已经定位到div标签 想要提取div中的a 只需在此填写a 而不用填写div > a） |

1. 列表排重（不可嵌套）

对于要爬取的数据进行去重。可根据url，标题等之前配置的属性进行去重。

|  |  |  |
| --- | --- | --- |
| 引擎选择（mongodb） |  | |
|  | 主机 | 主机地址 默认localhost |
| 端口 | 端口号 默认27017 |
| 用户（可选） | 用户名 |
| 密码（可选） | 密码 |
| 数据库名 | Mongo的数据库名称 |
| 集合名 | Mongo的collection名称 |
| 引擎选择（jdbc） |  | |
|  | 驱动 | Jdbc的驱动方式 |
| url | 数据库的地址 |
| 用户 | 用户名 |
| 密码 | 密码 |
| Table | 表名 |
| 引擎选择（redis） |  | |
|  | 主机 | Redis主机地址 |
| 端口 | 端口号 |
| 数据库索引 | 0 |
| 键 |  |
| 排重方式 | | MD5 |
| 属性字段名 | | 要排重的字段名 |
| 数据库映射字段名 | | 数据库中相对应的字段名 |

1. 提取列表（点击链接、创建列表、循环翻页）

提取网页中的列表元素

|  |  |
| --- | --- |
| 字段 | 字段名 |
| 提取列表（自定义方法） | 自定义方法的类名 |

1. 提取数据（静态）（不可嵌套）

可提取的信息包括该列表url页面上html所包含的全部信息。并可提取多个字段信息。如果详情页面是静态页面则采用静态的提取数据方式。

|  |  |  |
| --- | --- | --- |
| 标题（可选） | | 新闻标题 |
| 抽取字段 |  | |
|  | 字段名 | 抽取字段的名称 |
| 方式 | 定位字段元素的方式（Xpath，Regex，CSS，Common，Constant，Index） |
| 选择器 | 定位元素的路径（元素路径相对于上一层嵌套组件获得的元素路径 例如上一层已经定位到div标签 想要提取div中的a 只需在此填写a 而不用填写div > a） |
| 属性 | 元素中的属性 |
| 前缀检查（可选） | 判断抽取字段是否需要添加前缀 填入检查的前缀字符串 |
| 前缀（可选） | 依赖于前缀检查 如需要添加前缀 对于抽取字段添加前缀 |
| 后缀检查（可选） | 判断抽取字段是否需要添加后缀 填入检查的后缀字符串 |
| 后缀（可选） | 依赖于后缀检查 如需要添加后缀 对于抽取字段添加后缀 |
| 字符串截取（可选） | 截取表达式 以,分割 |
| 组号（可选） | 通过正则表达式捕获的组号 |
| 常量值（可选） | 常量 |
| 节点位置（可选） | 通过index方法获得的节点位置 从0开始 |
| 格式器（可选） | 可选time trim longtime |

1. 提取数据（动态）（不可嵌套）

可提取的信息包括该列表url页面上html所包含的全部信息。并可提取多个字段信息。如果详情页面是动态页面则采用动态的提取数据方式。

|  |  |  |
| --- | --- | --- |
| 抽取字段 |  | |
|  | 字段名 | 抽取字段的名称 |
| 方式 | 定位字段元素的方式（Id，Class，Css，TagName，Name，LinkText，ParitialLinkText，Xpath，JavaScript） |
| 选择器 | 定位元素的路径（元素路径相对于上一层嵌套组件获得的元素路径 例如上一层已经定位到div标签 想要提取div中的a 只需在此填写a 而不用填写div > a） |
| 属性 | 元素中的属性 |
| 前缀检查（可选） | 判断抽取字段是否需要添加前缀 填入检查的前缀字符串 |
| 前缀（可选） | 依赖于前缀检查 如需要添加前缀 对于抽取字段添加前缀 |
| 后缀检查（可选） | 判断抽取字段是否需要添加后缀 填入检查的后缀字符串 |
| 后缀（可选） | 依赖于后缀检查 如需要添加后缀 对于抽取字段添加后缀 |
| 字符串截取（可选） | 截取表达式 以,分割 |
| 组号（可选） | 通过正则表达式捕获的组号 |
| 常量值（可选） | 常量 |
| 节点位置（可选） | 通过index方法获得的节点位置 从0开始 |
| 格式器（可选） | 可选time trim longtime |
| 提取数据（自定义） | | 自定义方法的class名称 |

1. 数据存储（不可嵌套）

将爬取的字段数据存入数据库。

|  |  |  |
| --- | --- | --- |
| 引擎选择（mongodb） |  | |
|  | 主机 | 主机地址 默认localhost |
| 端口 | 端口号 默认27017 |
| 用户（可选） | 用户名 |
| 密码（可选） | 密码 |
| 数据库名 | Mongo的数据库名称 |
| 集合名 | Mongo的collection名称 |
| 引擎选择（jdbc） |  | |
|  | 驱动 | Jdbc的驱动方式 |
| url | 数据库的地址 |
| 用户 | 用户名 |
| 密码 | 密码 |
| Table | 表名 |
| 引擎选择（elasticsearch） |  | |
|  | 集群名 | Es集群名称 |
| 主机 | Es主机地址 |
| 端口 | Es端口号 |
| 数据库索引 |  |
| 类型 |  |
| 点击（自定义） | | 自定义方法的类名 |
| 属性字段名 | | 要存贮的字段名 |
| 数据库映射字段名 | | 数据库中相对应的字段名 |

1. 输入文本（不可嵌套）

|  |  |
| --- | --- |
| 方式 | 定位列表元素的方式（Id，Class，Css，TagName，Name，Xpath） |
| 选择器 | 定位元素的路径（元素路径相对于上一层嵌套组件获得的元素路径 例如上一层已经定位到div标签 想要提取div中的a 只需在此填写a 而不用填写div > a） |
| 脚本超时时间（script timout） | 默认5000毫秒 |
| 加载超时时间（load timeout） | 默认5000毫秒 |
| 等待超时时间（explicit wait） | 默认5000毫秒 |
| 输入文本 | 字符串 |

1. 页面滚动（不可嵌套）

|  |  |
| --- | --- |
| 页面滚动脚本 | 页面滚动js脚本 |
| 参数 | 参数列表 |
| 脚本超时时间（script timout） | 默认5000毫秒 |
| 加载超时时间（load timeout） | 默认5000毫秒 |
| 等待超时时间（explicit wait） | 默认5000毫秒 |

1. **样例**

### 3.1 证券之星(静态爬虫)

#### 3.1.1 打开网址

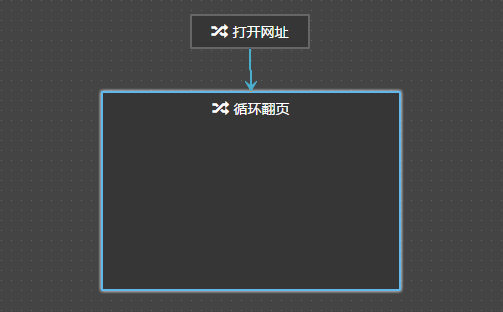


3.1.1流程图

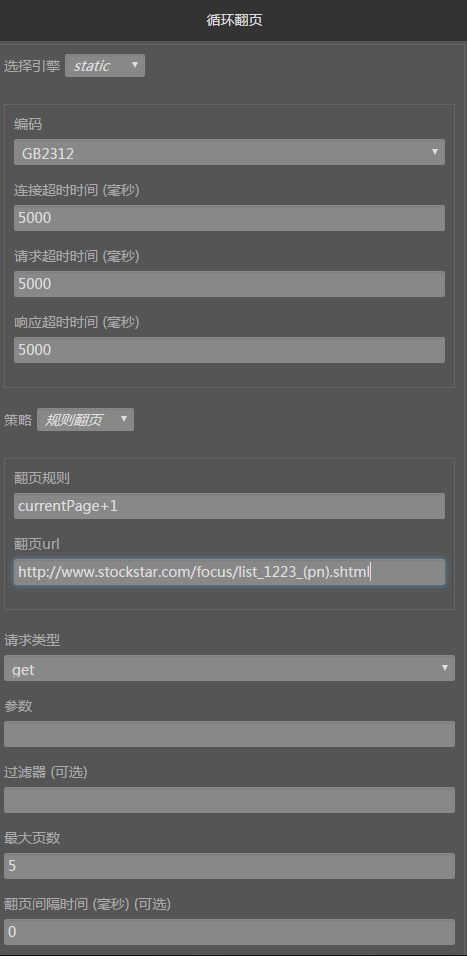


3.1.2 配置属性图

#### 3.1.2 循环翻页



3.1.3 流程图



3.1.4 配置属性图

#### 3.1.3 创建列表

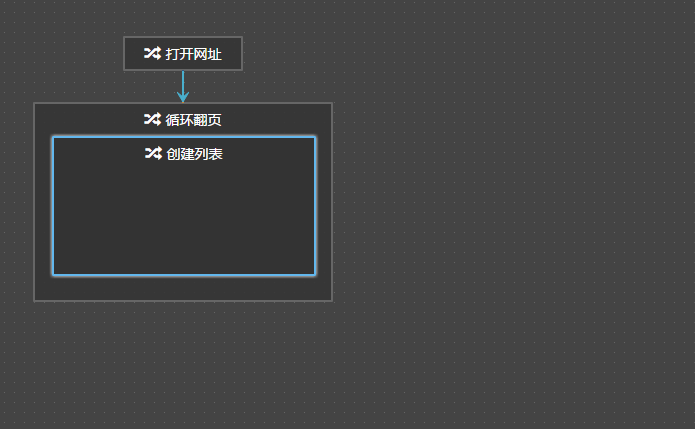


图3.1.5 流程图

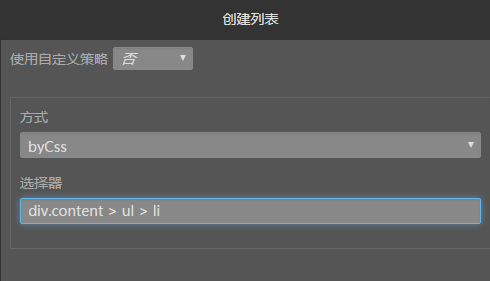


图3.1.6 配置属性图

#### 3.1.4 提取数据 （提取列表信息）

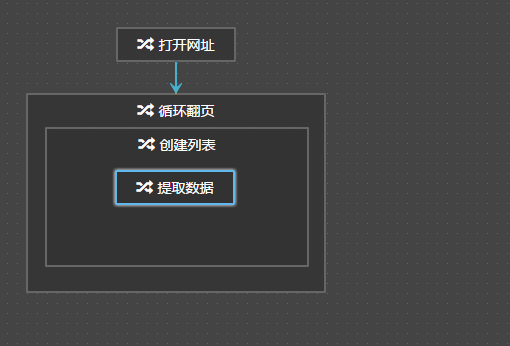


图3.1.7 流程图

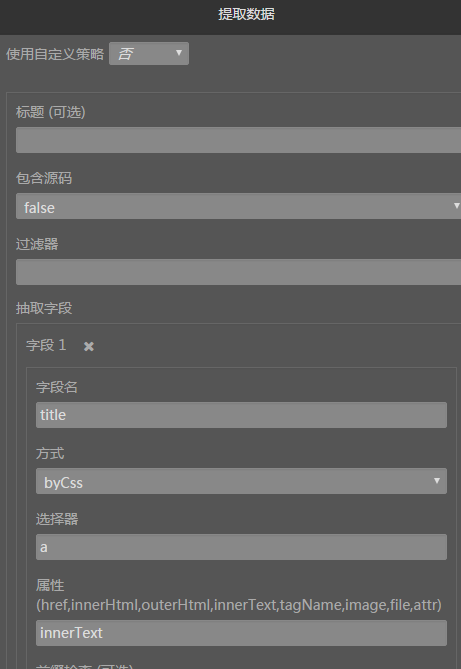


图3.1.8 配置属性图

#### 3.1.5 列表排重

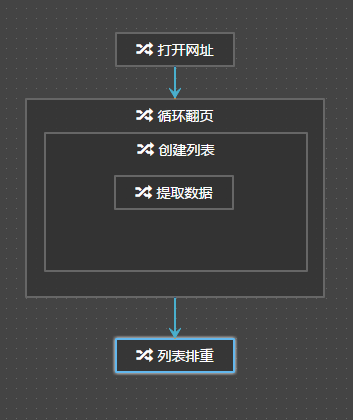


图3.1.9 流程图



图3.1.10 配置属性图

#### 3.1.6 处理列表

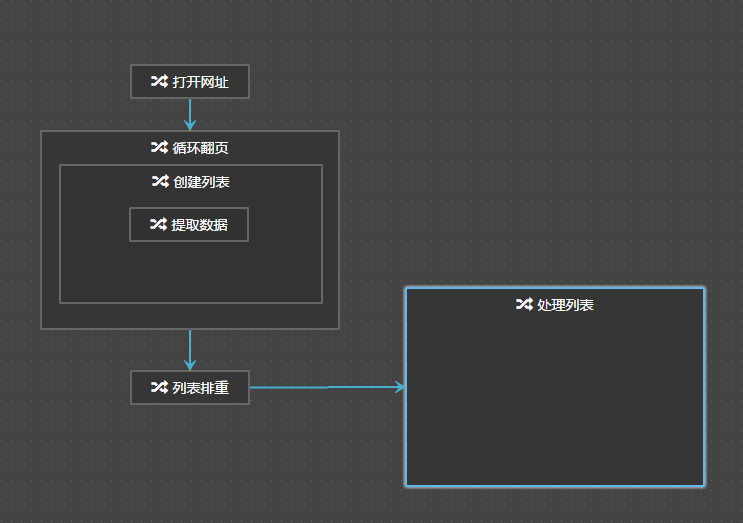


图3.1.11 流程图

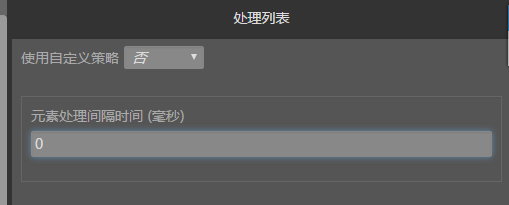


图3.1.12 配置属性图

#### 3.1.7 点击链接

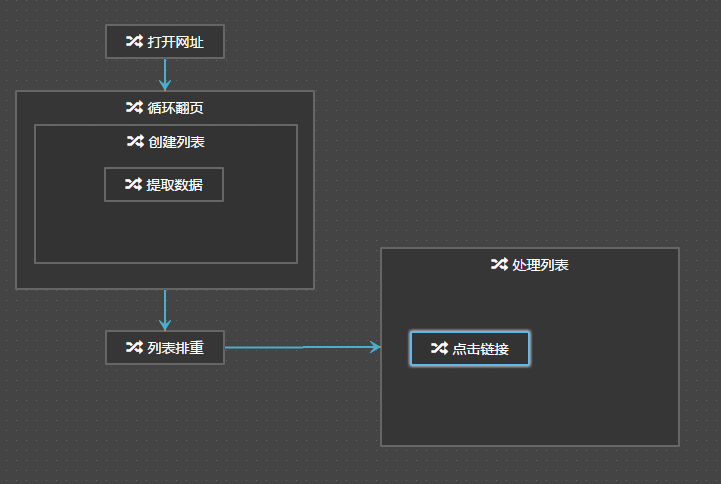


图3.1.13 流程图

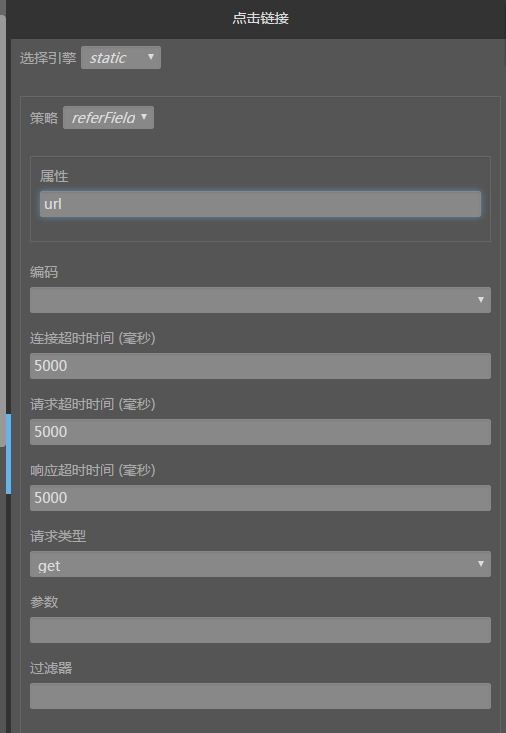


图3.1.14 配置属性图

#### 3.1.8 提取数据（提取页面详细数据）

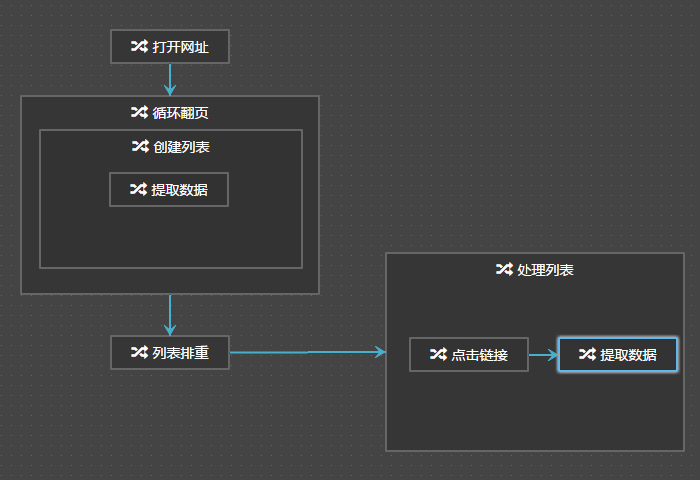


图3.1.15 流程图



图3.1.16 配置属性图

#### 3.1.9 数据存储

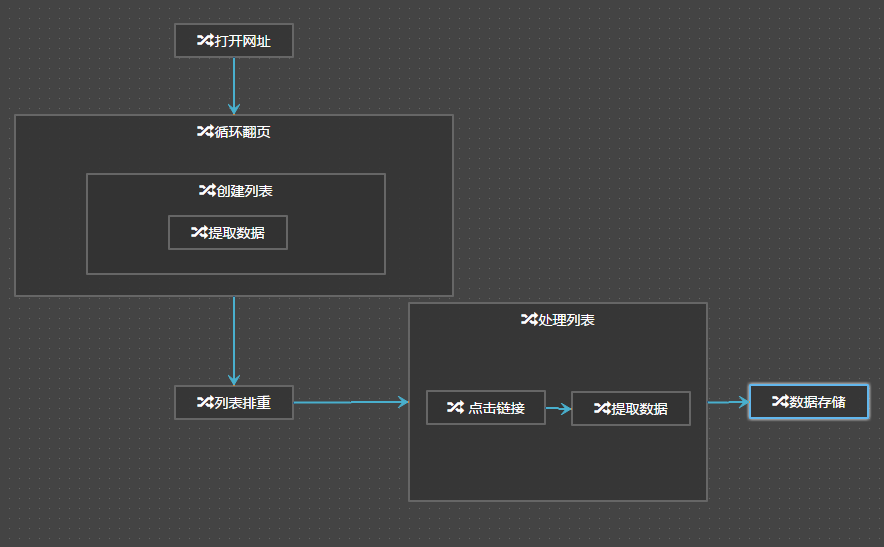


图3.1.17 流程图

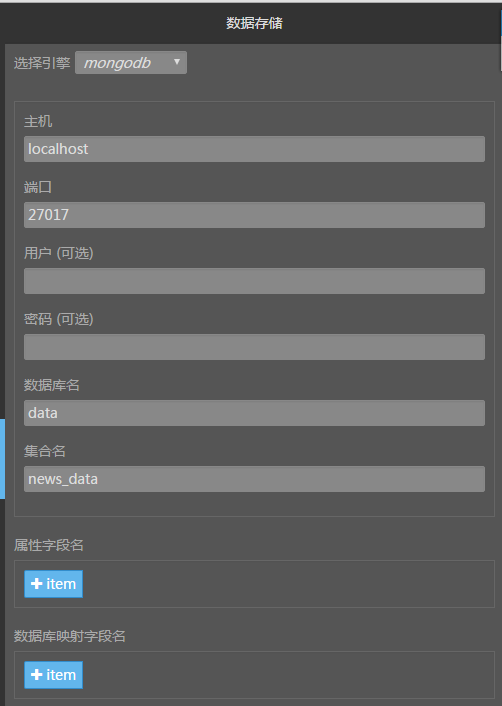


图3.1.18 配置属性图

### 3.2 今日头条(动态爬虫)

#### 3.2.1 打开网址

通过动态引擎打开动态网页。



图3.2.1 流程图

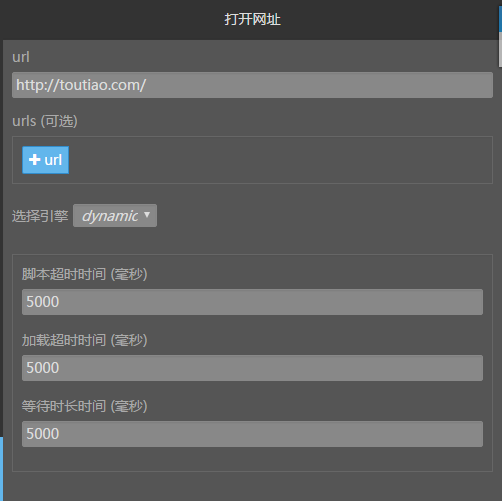


图3.2.2 配置属性图

#### 3.2.2 循环翻页

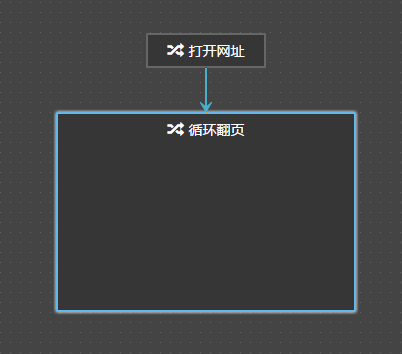


图3.2.3 流程图

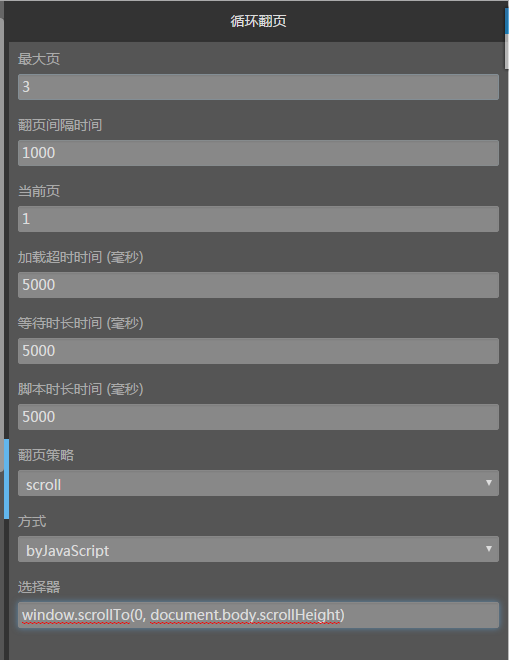


图3.2.4 配置属性图

#### 3.2.3 创建列表

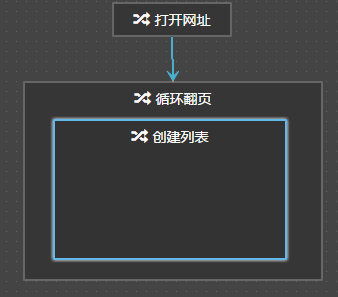


图3.2.5 流程图

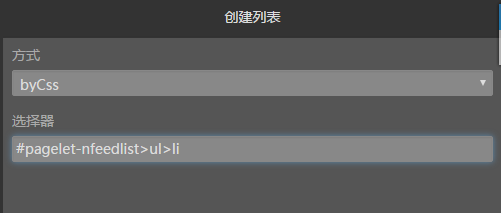


图3.2.6 配置属性图

#### 3.2.4 提取数据

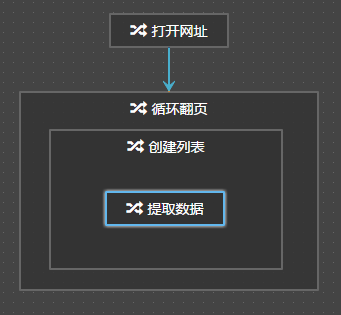


图3.2.7 流程图

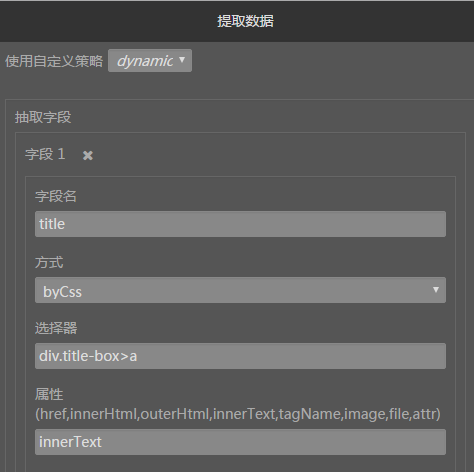


图3.2.8 配置属性图

#### 3.2.5 列表排重

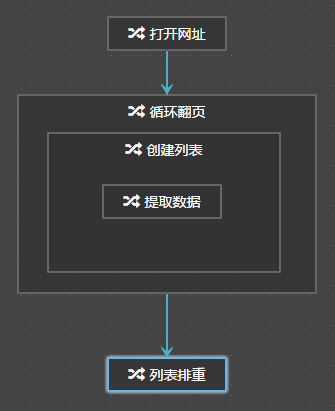


图3.2.9 流程图

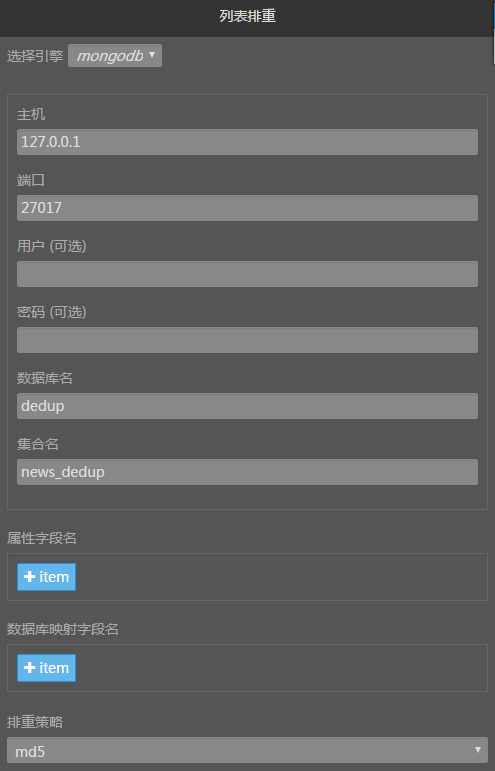


图3.2.10 配置属性图

#### 3.2.6 处理列表

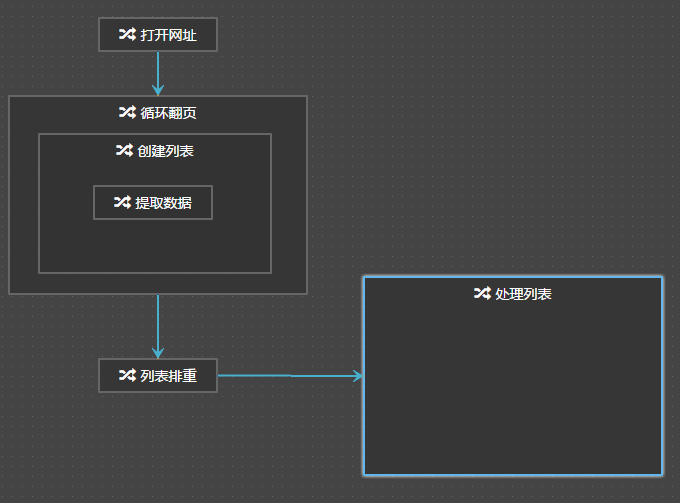


图3.2.11 流程图

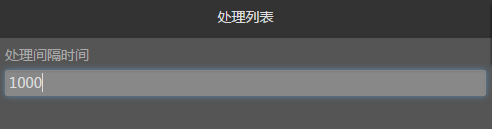


图3.2.12 配置属性图

#### 3.2.7 打开网址

打开网址依赖于之前提取的url字段值 引用属性以$开头

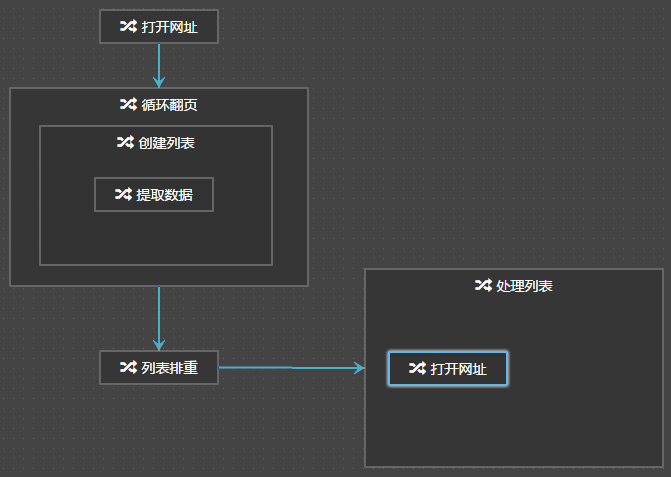


图3.2.13 流程图

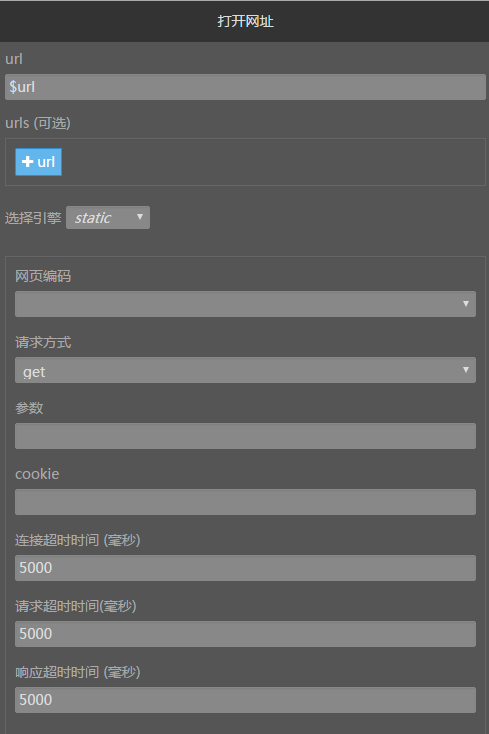


图3.2.14 配置属性图

#### 3.2.8 提取数据

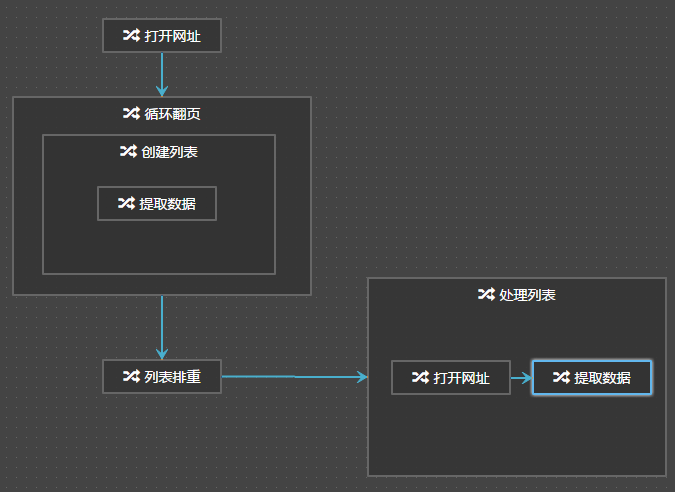


图3.2.15 流程图



图3.2.16 配置属性图

#### 3.2.9 数据存储

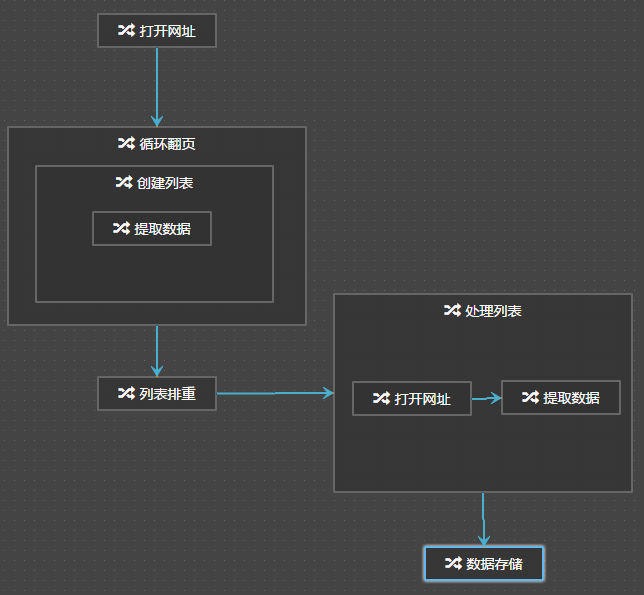


图3.2.17 流程图

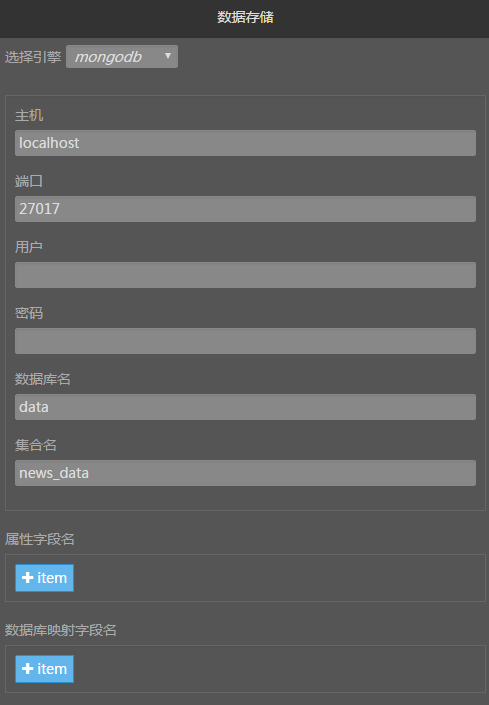


图3.2.18 配置属性图

1. **附录**

周期表达式主要是给定点采集使用，也就是说采集器会在一个固定的时间点上启动。

表达式配置：类似linux中的crontab

不过比crontab多一个秒

具体表达式格式：

Seconds

Minutes

Hours

Day of month

Month

Day of week

说明：

Seconds表达式：支持数字0-59 支持的特殊字符,\*-/

Minutes 表达式：支持数字0-59 支持的特殊字符,\*-/

Hours 表达式：支持数字0-23 支持的特殊字符,\*-/

Day of month表达式：支持数字1-31 支持的特殊字符,\*-/?

Month表达式：支持数字1-7 支持的特殊字符,\*-/

Day of week表达式：支持数字0-59 支持的特殊字符,\*-/?

注意：

1.只有Day of month和Day of week支持?这个特殊字符，意思是说，当他们两个我只要满足一个就行了，然后另一个用?代替，表示不管另一个。

2.都支持的特殊字符

,表示多个单值 枚举 比如Seconds 上写 1,2,3 意思是：每分钟的1、2、3秒会启动调度

\*表示每一个值 比如Seconds 上写 \* 意思是：每分钟的每一秒都会启动调度

-表示值区间 比如Seconds 上写 1-3 意思是：每分钟的1、2、3秒会启动调度， 和写1,2,3 效果一样

/表示每隔多久 比如Seconds 上写1/3 意思是：每隔3秒启动一次调度 并且第一次启动在从最近的1秒开始计算

每一个表达式项之间用英文空格区分

举例：

30/1 \* \* \* \* ?

表示每隔一秒启动一次调度，第一次启动在最近的30秒的时候。