**监控工具开发文档**

1. **项目环境**

系统：Window

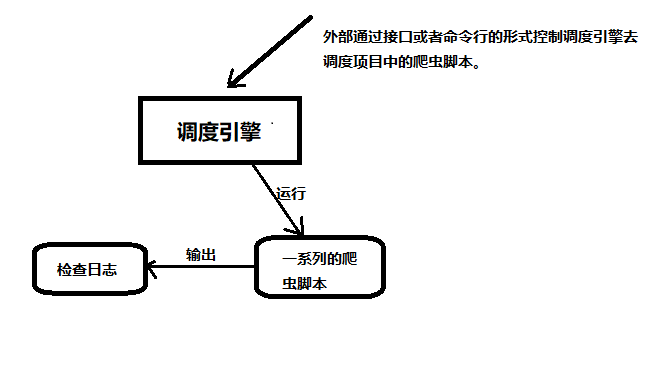
程序：python3.6

框架：scrapy1.5

1. **监控的原理**

通过为每个采集点写一个爬虫的脚本，用脚本去采集点解析首页和数十个细缆页，达到监控的目的。首页解析其包含的细缆链接，细缆页解析每篇新闻所需的字段。 解析的方式有xpath路径，css样式，正则。为每个字段应用一个详细的多级解析规则，这样原网页与该字段相关的网页结构一稍微变化，爬虫脚本就会提示该网页的结构变化。每个采集点都有其相应的爬虫脚本。最终程序会把所有采集点的监控日志汇总到一个当天的日志文件。监控的频率暂定为每天跑一次。

1. **项目的结构**

****

1. 调度引擎为一个脚本汇总的框架接口，控制项目中爬虫脚本的运行。
2. 爬虫脚本：每个采集点有其相应的一个脚本，脚本的命名不可重复，建议以采集点的全拼进行命名。
3. 检查日志：所有采集点的检查日志汇总。

**4．目录结构**

NewsCrawl项目的名字。

NewsCrawl\NewsCrawl\items.py:定义了要取什么字段

NewsCrawl\NewsCrawl\spiders\采集点监控脚本的文件夹

NewsCrawl\result\项目的日志目录，按日期来给日志命名，日志记录了采集点的监控情况

NewsCrawl\NewsCrawl\pipelines.py:对采集点是否正常进行判断，并输出判断结果到文件。

NewsCrawl\NewsCrawl\settings:每一个爬虫脚本，结构如模板所所示，项目中的设置为项目的通用设置,若每个spider需要的个性化的设置，可在类的属性里添加针对每个爬虫的个性化设置，代理，请求头，下载延迟，请求重试次数等等设置，个性化设置会覆盖项目的通用设置。

**5.日志部分**

a.日志目录：NewsCrawl\result\

b.日志的名称为当天的日期，日志字段的分隔符为 ^,日志有13个字段，可用excel打开进行筛选错误，无采集点去重，重复监控会出现重复的结果。

c.字段说明:

网站名：网站的名字

网址：网站的网址

开始时间：开始检测这个采集点的时间

用时：该采集点的监控用时

首页检擦：可能结果 1或0或空 1:有错，0:正常，空:该页无响应

标题：同上

正文：同上

来源：同上

作者：同上

时间：同上

引题：同上

副题：同上

错误总数：以上七个字段错误的累加值，等于0：表示该采集点正常，大于0：采 集点结构改变，无响应：该采集点无响应

**6.爬虫脚本的书写规范**

脚本提取字段的xpath或者css的规则写的越详细越好，这样原网站一稍稍变动，检测就能报错，起到更好的检测作用

**1**.类继承 CrawlSpider

xxxClass(CrawlSpider)

**2**.类属性

*name:* 采集点的名字

*start\_urls:* 始爬虫的url列表

rules : 从首页中提取概览页的链接，参数灵活，有多种提取方式，xptah，正则，css，链接排除....

**3.**类方法

*parse\_start\_url:* 从首页检查url提取是否正常，并return提取的内容

*parse\_item:* 从细缆页从提取具体的字段，并return提取的结果

**7.项目的进度**

**a.外部的引擎结构规划大致完成，正在进行外部框架的开发。**

**b.主要的难点**

**工作量：**采集点的数量很多，要分别为每个采集点开发一个脚本。每个采集点脚本的书写规范，字段的提取规则决定了监控的准备性。

**外部框框架引擎的稳定性：**在引擎需要调度很多脚本的情况下，项目依然能够有条不紊的运行，目前尚未有很多的采集点脚本，稳定性待定。