**目标系统任务书**

专 业 名 称 ：计算机科学与技术

课 程 名 称 ：软件工程

指 导 教 师 ：刘纪平

小 组 成 员 ：

张欣瑶 2020302111370

卞江畔 2020302111037

朱蓉蓉 2020302111062

高 璇 2020302111076

二○二三年三月

# **1.目标系统名称**

Scrapy爬虫框架

# 2.基本情况介绍

# 3.目标和要求

# 4.特征和影响

# 5.验收建议

**5.1优缺点分析**

scrapy框架，将网页采集的通用功能，集成到各个模块中，留出自己定义的部分，它将程序员从繁冗的流程式重复劳动中解放出来，简单的网页爬虫的重点，就集中在应对反爬，大规模爬取和高效稳定的爬取这几个方面。

优点如下：

1. scrapy 是异步的，可以灵活调节并发量

2.采取可读性更强的 xpath 代替正则，速度快

3.写 middleware,方便写一些统一的过滤器

4.同时在不同的 url 上爬行

5.支持 shell 方式，方便独立调试

6.通过管道的方式存入数据库，灵活，可保存为多种形式

但是也存在一些不近如人意的地方，例如：

1. 无法用它完成分布式爬取
2. 自身去重效果差，消耗内存，且不能持久化
3. 对于需要执行js才能获取数据的网页，爱莫能助
4. 兼容了下载图片与视频，但是可用性比较差
5. 自身扩展的log模块，不好用，经常需要自定义
6. 基于 twisted 框架，运行中的 exception 是不会干掉 reactor（反应器），并且异步框架出错后 是不会停掉其他任务的，数据出错后难以察觉，预警系统简单的使用邮件，很不友好

**5.2希望达成的目标**

Scrapy 常应用在包括数据挖掘，信息处理或存储历史数据等一系列的程序中。通常我们可以很简单的通过 Scrapy 框架实现一个爬虫，抓取指定网站的内容或图片。

**5.3验收方式**

1.创建项目scrapy project xxx

2.制作sprider scrapy genspider xxx "[http://www.xxx.com](http://www.xxx.com/)"

3.编写Item.py (明确需要提取的数据)

4.编写爬虫文件（spiders/xxx.py）处理请求和响应，以及提取数据(yield item)

5.编写管道文件（pipelines.py）处理spider返回的item数据，比如本地持久化存储等。

6.编写settings.py，启动管道组件 ITEM\_PIPELINES={}，以及其他相关设置。

7.执行爬虫 scrapy crawl xxx