Scrapy爬虫框架

Scrapy爬虫框架

- Scrapy架构流程
- Scrapy爬虫步骤

1-1. Scrapy架构流程

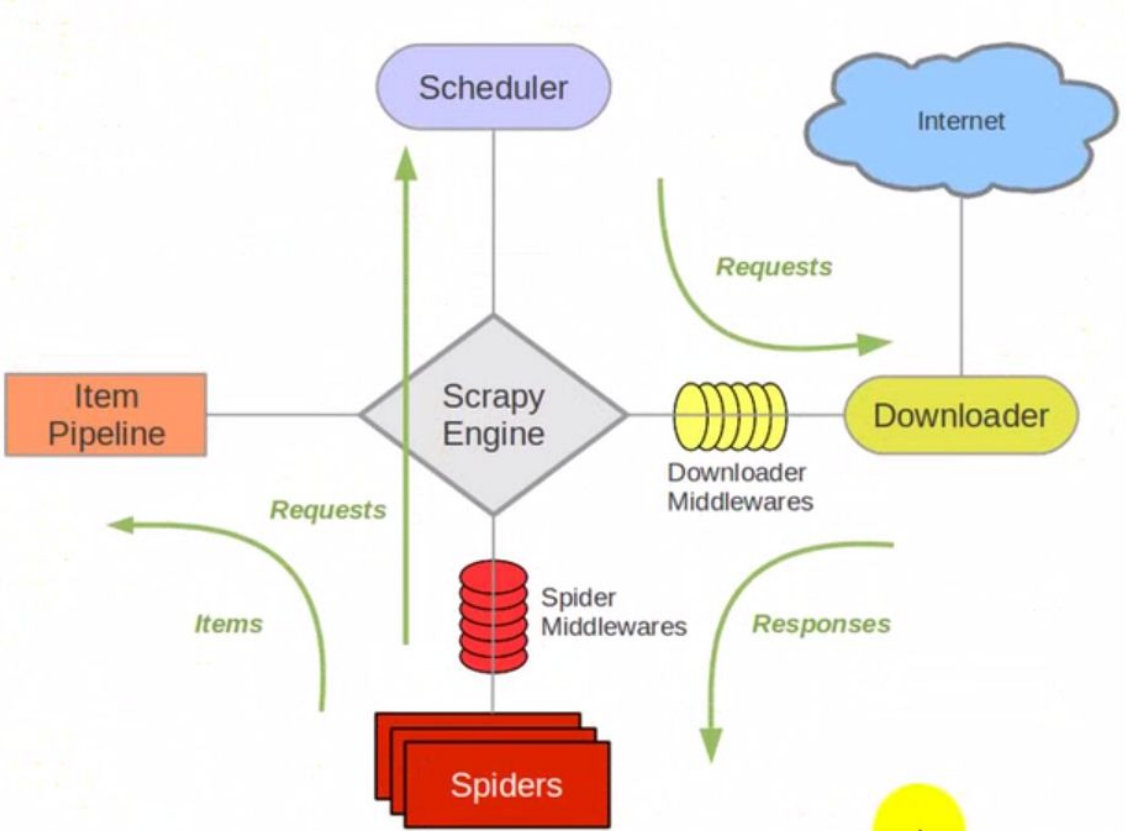
- Scrapy，Python开发的一个快速、高层次的屏幕抓取和web抓取框架，用于抓取web站点并从页面中提取结构化的数据。
- Scrapy吸引人的地方在于它是一个框架，任何人都可以根据需求方便的修改。它也提供了多种类型爬虫的基类，如BaseSpider、sitemap爬虫等，最新版本又提供了web2.0爬虫的支持。
- Scrap，是碎片的意思，这个Python的爬虫框架叫Scrapy。

1-1. Scrapy架构流程

优势:

- 用户只需要定制开发几个模块， 就可以轻松实现爬虫, 用来抓取网页内容和图片， 非常方便;
- Scrapy使用了Twisted异步网络框架来处理网络通讯， 加快网页下载速度， 不需要自己实现异步框架和多线程等， 并且包含了各种中间件接口， 灵活完成各种需求

1-1. Scrapy架构流程



架构图(绿线是数据流向)

1-1. Scrapy架构流程

Scrapy主要包括了以下组件：

- 引擎(Scrapy): 用来处理整个系统的数据流, 触发事务(框架核心)
- 调度器(Scheduler): 用来接受引擎发过来的请求, 压入队列中, 并在引擎再次请求的时候返回. 可以想像成一个URL（抓取网页的网址或者说是链接）的优先队列, 由它来决定下一个要抓取的网址是什么, 同时去除重复的网址
- 下载器(Downloader): 用于下载网页内容, 并将网页内容返回给蜘蛛(Scrapy下载器是建立在twisted这个高效的异步模型上的)

1-1. Scrapy架构流程

Scrapy主要包括了以下组件：

- 爬虫(Spiders): 爬虫是主要干活的，用于从特定的网页中提取自己需要的信息，即所谓的实体(Item)。用户也可以从中提取出链接,让Scrapy继续抓取下一个页面
- 项目管道(Pipeline):负责处理爬虫从网页中抽取的实体，主要的功能是持久化实体、验证实体的有效性、清除不需要的信息。当页面被爬虫解析后，将被发送到项目管道，并经过几个特定的次序处理数据。

1-1. Scrapy架构流程

Scrapy主要包括了以下组件：

- 下载器中间件(Downloader Middlewares): 位于Scrapy引擎和下载器之间的框架，主要是处理Scrapy引擎与下载器之间的请求及响应。
- 爬虫中间件(Spider Middlewares):介于Scrapy引擎和爬虫之间的框架，主要工作是处理蜘蛛的响应输入和请求输出
- 调度中间件(Scheduler Middewares): 介于Scrapy引擎和调度之间的中间件，从Scrapy引擎发送到调度的请求和响应。

1-1. Scrapy架构流程

代码写好，程序开始运行...

1. 引擎 ：Hi！ Spider ，你要处理哪一个网站？
2. Spider ：老大要我处理xxxx.com。
3. 引擎 ：你把第一个需要处理的URL给我吧。
4. Spider ：给你，第一个URL是xxxxxxx.com。
5. 引擎 ：Hi！ 调度器 ，我这有request请求你帮我排序入队一下。
6. 调度器 ：好的，正在处理你等一下。

1-1. Scrapy架构流程

7. 引擎 ：Hi！ 调度器 ，把你处理好的request请求给我。
8. 调度器 ：给你，这是我处理好的request
9. 引擎 ：Hi！下载器，你按照老大的 下载中间件 的设置帮我下载一下这个request请求
10. 下载器 ：好的！给你，这是下载好的东西。（如果失败：sorry，这个request下载失败了。然后 引擎 告诉 调度器 ，这个request下载失败了，你记录一下，我们待会儿再下载）
11. 引擎 ：Hi！ Spider ，这是下载好的东西，并且已经按照老大的 下载中间件 处理过了，你自己处理一下（注意！这儿responses默认是交给 def parse() 这个函数处理的）
12. Spider ：（处理完毕数据之后对于需要跟进的URL），Hi！ 引擎 ，我这里有两个结果，这个是我需要跟进的URL，还有这个是我获取到的Item数据。
13. 引擎 ：Hi ！ 管道 我这儿有个item你帮我处理一下！ 调度器 ！这是需要跟进URL你帮我处理下。然后从第四步开始循环，直到获取完老大需要全部信息。
14. 管道``调度器 ：好的，现在就做！

只有当调度器中不存在任何request时， 整个程序才会停止。(注:对于下载失败的URL，Scrapy也会重新下载.)

1-2. Scrapy爬虫步骤

- 新建项目(scrapy startproject xxx):
  - 新建一个新的爬虫项目;
- 明确目标(编写item.py)
  - 明确你要抓取的目标;
- 制作爬虫(spiders/xxspider.py)
  - 制作爬虫， 开始爬取网页;
- 存储爬虫(pipelines.py)
  - 设置管道存储爬取内容;