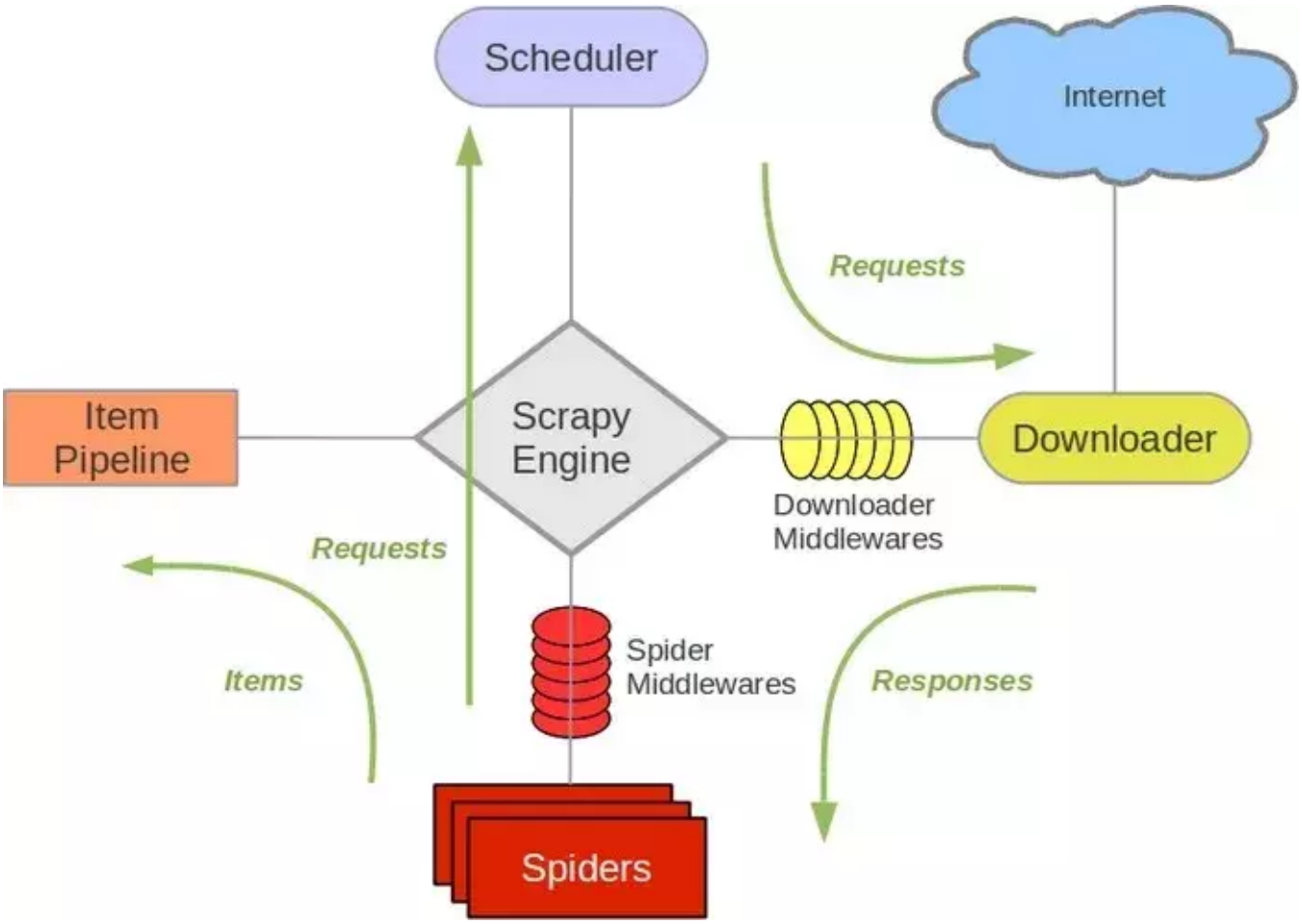
崔庆才丨静觅
2018年05月10日 阅读 573

Scrapy框架的使用之Spider Middleware的用法

Spider Middleware是介入到Scrapy的Spider处理机制的钩子框架。我们首先来看看它的架构，如下图所示。



当Downloader生成Response之后，Response会被发送给Spider，在发送给Spider之前，Response会首先经过Spider Middleware处理，当Spider处理生成Item和Request之后，Item和Request还会经过Spider Middleware的处理。

Spider Middleware有如下三个作用。

- 我们可以在Downloader生成的Response发送给Spider之前，也就是在Response发送给Spider之前对Response进行处理。

- 我们可以在Spider生成的Item发送给Item Pipeline之前，也就是在Item发送给Item Pipeline之前对Item进行处理。

一、使用说明

需要说明的是，Scrapy其实已经提供了许多Spider Middleware，它们被 `SPIDER_MIDDLEWARES_BASE` 这个变量所定义。

`SPIDER_MIDDLEWARES_BASE` 变量的内容如下：

```
{
    'scrapy.spidermiddlewares.httperror.HttpErrorMiddleware': 50,
    'scrapy.spidermiddlewares.offsite.OffsiteMiddleware': 500,
    'scrapy.spidermiddlewares.referer.RefererMiddleware': 700,
    'scrapy.spidermiddlewares.urllength.UrlLengthMiddleware': 800,
    'scrapy.spidermiddlewares.depth.DepthMiddleware': 900,
}
```

和Downloader Middleware一样，Spider Middleware首先加入到 `SPIDER_MIDDLEWARES` 设置中，该设置会和Scrapy中 `SPIDER_MIDDLEWARES_BASE` 定义的Spider Middleware合并。然后根据键值的数字优先级排序，得到一个有序列表。第一个Middleware是最靠近引擎的，最后一个Middleware是最靠近Spider的。

二、核心方法

Scrapy内置的Spider Middleware为Scrapy提供了基础的功能。如果我们想要扩展其功能，只需要实现某几个方法即可。

每个Spider Middleware都定义了以下一个或多个方法的类，核心方法有如下4个。

- `process_spider_input(response, spider)` 。
- `process_spider_output(response, result, spider)` 。
- `process_spider_exception(response, exception, spider)` 。
- `process_start_requests(start_requests, spider)` 。

1. process_spider_input(response, spider)

当Response被Spider Middleware处理时， `process_spider_input()` 方法被调用。

`process_spider_input()` 方法的参数有如下两个。

- `response` ，是Response对象，即被处理的Response。
- `spider` ，是Spider对象，即该Response对应的Spider。

`process_spider_input()` 应该返回None或者抛出一个异常。

- 如果它返回None，Scrapy将会继续处理该Response，调用所有其他的Spider Middleware，直到Spider处理该Response。
- 如果它抛出一个异常，Scrapy将不会调用任何其他Spider Middleware的 `process_spider_input()` 方法，而调用Request的 `errback()` 方法。 `errback` 的输出将会被重新输入到中间件中，使用 `process_spider_output()` 方法来处理，当其抛出异常时则调用 `process_spider_exception()` 来处理。

2. process_spider_output(response, result, spider)

当Spider处理Response返回结果时， `process_spider_output()` 方法被调用。

`process_spider_output()` 方法的参数有如下三个。

- `response` ，是Response对象，即生成该输出的Response。
- `result` ，包含Request或Item对象的可迭代对象，即Spider返回的结果。
- `spider` ，是Spider对象，即其结果对应的Spider。

`process_spider_output()` 必须返回包含Request或Item对象的可迭代对象。

3. process_spider_exception(response, exception, spider)

`process_spider_exception()` 方法的参数有如下三个。

- `response` ，是Response对象，即异常被抛出时被处理的Response。
- `exception` ，是Exception对象，即被抛出的异常。
- `spider` ，是Spider对象，即抛出该异常的Spider。

`process_spider_exception()` 必须要么返回 `None` ，要么返回一个包含Response或Item对象的可迭代对象。

- 如果它返回 `None` ，Scrapy将继续处理该异常，调用其他Spider Middleware中的 `process_spider_exception()` 方法，直到所有Spider Middleware都被调用。
- 如果它返回一个可迭代对象，则其他Spider Middleware的 `process_spider_output()` 方法被调用，其他的 `process_spider_exception()` 不会被调用。

4. process_start_requests(start_requests, spider)

`process_start_requests()` 方法以Spider启动的Request为参数被调用，执行的过程类似于 `process_spider_output()` ，只不过它没有相关联的Response，并且必须返回Request。

`process_start_requests()` 方法的参数有如下两个。

- `start_requests` ，是包含Request的可迭代对象，即Start Requests。
- `spider` ，是Spider对象，即Start Requests所属的Spider。

`process_start_requests()` 必须返回另一个包含Request对象的可迭代对象。

三、结语

本节介绍了Spider Middleware的基本原理和自定义Spider Middleware的方法。Spider Middleware使用的频率不如Downloader Middleware的高，在必要的情况下它可以用来方便数据的处理。

本资源首发于崔庆才的个人博客静觅： Python3网络爬虫开发实战教程 | 静觅