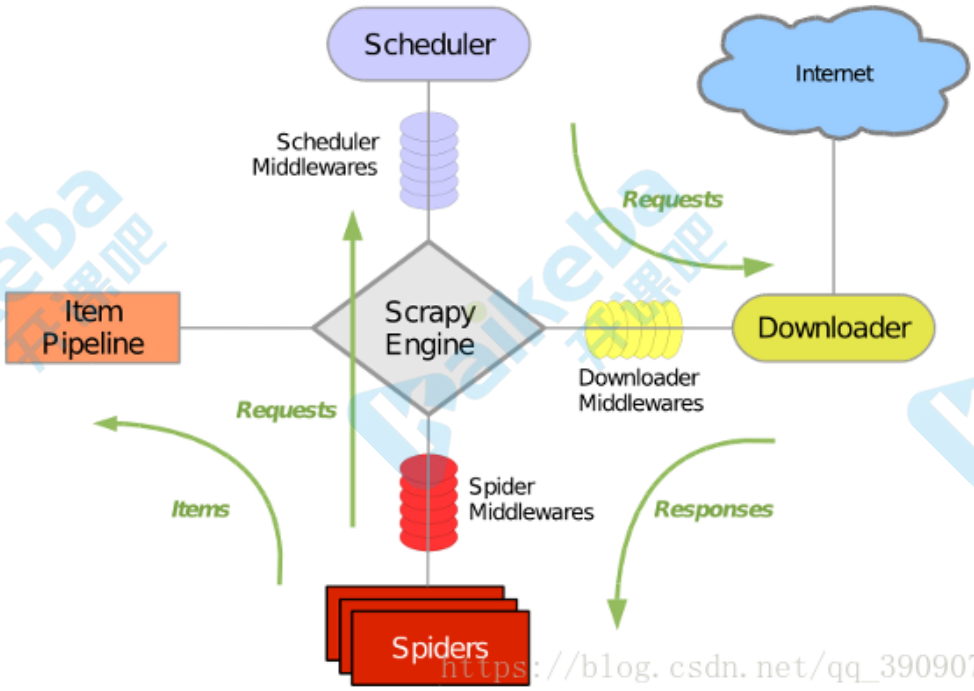
Scrapy介绍（一）

Scrapy介绍

Scrapy一个开源和协作的框架，其最初是为了页面抓取（更确切来说，网络抓取）所设计的，使用它可以快速、简单、可扩展的方式从网站中提取所需的数据。但目前Scrapy的用途十分广泛，可用于如挖掘、监测和自动化测试等领域，也可以应用在API所返回的数据（例如Amazon Associates Web Services）或者通用的网络爬虫。

Scrapy是基于twisted框架开发而来，twisted是一个流行的事件驱动的python网络框架。因此Scrapy使用了一种非阻塞（又名异步）的代码来实现并发。整体架构大致如下



Scrapy数据流是由执行的核心引擎（engine）控制，流程是这样的：

1.引擎打开一个网站（open adomain）,找到处理该网站的Spider并向该spider请求第一个要抓取的URL（s）。

2.引擎从Spider中获取到第一个要抓取的URL并在调度器（Scheduler）以Request调度。

3.引擎向调度器请求下一个要爬取的URL。

4.调度器返回下一个要抓取的URL给引擎，引擎将URL通过下载中间件（请求（request）方向）转发给下载器（Downloader）.

5.一旦页面下载完毕，下载器生成一个该页面的Response,并将其通过下载中间件（返回（response）方向）发送给引擎。

6.引擎从下载器中接收到Response并通过Spider中间件（输入方向）发送给Spider处理。

7.Spider处理Response并返回爬取到的Item给Item Pipeline,将（Spider返回的）Request给调度器。

8.引擎将（Spider返回的）爬取的Item给Item Pipeline,将（Spider返回的）Request给调度器。

9.（从第二步）重复直到调度器中没有更多地request,引擎关闭该网站。

Scrapy主要包括了以下组件：

1.爬虫引擎（engine）：爬虫引擎负责控制各个组件之间的数据流，当某些操作触发事件后都是通过engine来处理

2.调度器：调度接收来engine的请求并将请求请求放入队列中，并通过事件返回给engine

3.下载器：通过engine请求下载网络数据并将结果响应给engine

4.spider:Spider发出请求，并处理engine返回给它下载器响应数据，以items和规则内的数据请求（urls）返回给engine

5.管道数目（item pipeline）:负责处理engine返回spider解析后的数据，并且将数据持久化，例如将数据存入数据库或者文件

6.下载中间件：下载中间件是engine和下载器交互组件，以钩子（插件）的形式存在，可以代替接收请求、处理数据的下载以及将结果响应给engine

7.spider中间件：spider中间件是engine和spider之间的交互组件，以钩子（插件）的形式存在，可以代替处理response以及返回给engine items及新的请求集