3.2Scrapy框架

**3.2.1Scrapy框架简介**

Scrapy是一个为了爬取网站数据，提取结构性数据而编写的应用框架。 可以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。Scrapy是用Python开发的一个开源的Web爬虫框架，可用于快速抓取Web站点并从页面中高效提取结构化的数据。Scrapy可广泛应用于数据挖掘、监测和自动化测试等方面，提供了多种类型爬虫的基类，如BaseSpider,SitemapSpider等。

其最初是为了页面抓取 (更确切来说, 网络抓取 )所设计的， 也可以应用在获取API所返回的数据(例如 Amazon Associates Web Services ) 或者通用的网络爬虫。Scrapy用途广泛，可以用于数据挖掘、监测和自动化测试。

Scrapy 使用了 Twisted异步网络库来处理网络通讯。整体架构大致如下：

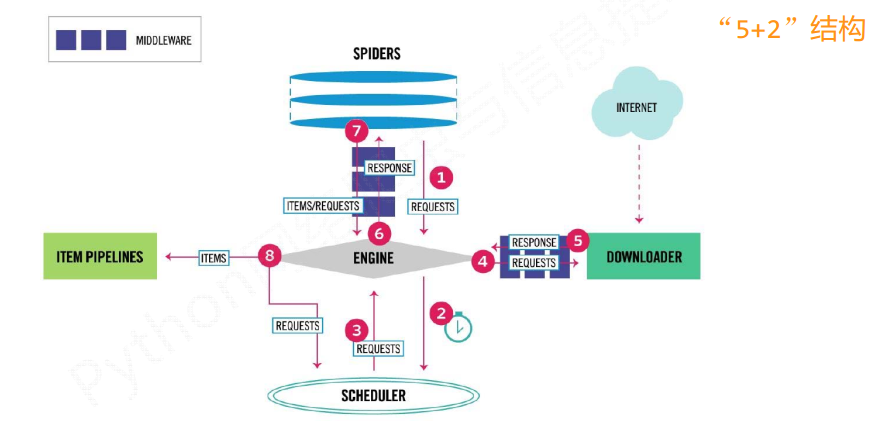


Scrapy框架组件：

如上图所示Scrapy框架包括了一下的：

* 引擎(Scrapy): 用来处理整个系统的数据流处理, 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)
* 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。
* 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。

**3.2.2Scrapy框架解析**：

Scrapy框架是采用了5+2结构主要有三条主要的数据流路径

第一条路径：

1) Engine通过中间件获得了Spiders发送的请求, 这个请求叫做requests, 相当于是一个url

2) Engine再转发给scheduler, scheduler主要负责对爬取请求进行调度

第二条路径：

3) 从Scheduler获得下一个需要爬取的请求, 这是一个真实的请求

4) Engine获得这个请求之后, 通过中间件, 将请求给 Downloader模块

5) Downloader爬取请求中相关的网页, 并将爬取的内容封装成一个对象, 这个对象叫response(响应)

6) Engine再通过中间件将response返回给Spiders

第三条路径：

7) Spiders处理从Downloader获得的响应, 处理之后会产生两个数据类型, 一个是items(爬取项), 另一个是新的requests

8) Engine接受到这两数据之后, 将items发送给Item Pipelines, requests发送给Scheduler进行调度

整个框架的入口是Spiders, 出口是Item Pipelines其中Engine, Scheduler, 和Downloader是已经写好的, 不需要实现Spiders和Item Pipelines是需要编写的, 但是里面有既定的代码框架, 所以情况是要对某东西进行修改, 这种情况一般叫配置

**3.2.3Scrapy创建**

1. Scrapy运行流程大概如下：

首先，引擎从调度器中取出一个链接(URL)用于接下来的抓取；

引擎把URL封装成一个请求(Request)传给下载器，下载器把资源下载下来，并封装成应答包(Response)；

然后，爬虫解析Response；

若是解析出实体（Item）,则交给实体管道进行进一步的处理；

若是解析出的是链接（URL）,则把URL交给Scheduler等待抓取；

1. Scrapy Tutorial

安装命令在IDE(pycharm中):pip install Scrapy

在抓取之前要新建一个项目:

CMD下到相应的目录中去输入命令:

scrapy startproject Sina\_spider

就会创建如下的文件夹结构:

├── scrapy.cfg

└── Sina\_spider

   ├── \_\_init\_\_.py

   ├── items.py

   ├── pipelines.py

   ├── settings.py

   └── spiders

       └── \_\_init\_\_.py

文件主要是:

* scrapy.cfg: 项目配置文件
* Sina\_spider/: 项目python模块, 之后您将在此加入代码
* Sina\_spider/items.py: 项目items文件
* Sina\_spider/pipelines.py: 项目管道文件
* Sina\_spider/settings.py: 项目配置文件
* Sina\_spider/spiders: 放置spider的目录