**Python爬虫(六)--Scrapy框架学习**

<http://www.jianshu.com/p/078ad2067419>

本文希望达到以下目标:

1、简要介绍Scarpy

2、阅读官网入门文档并实现文档中的范例

3、使用Scarpy优豆瓣爬虫的抓取

4、制定下一步学习目标

**1. Scrapy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。

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主要包括了以下组件：

* 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。

Scrapy运行流程大概如下：

1、首先，引擎从调度器中取出一个链接(URL)用于接下来的抓取

2、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)

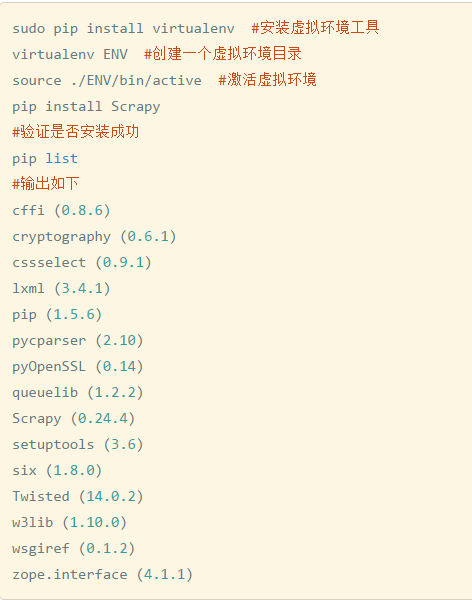
3、然后，爬虫解析Response

4、若是解析出实体（Item）,则交给实体管道进行进一步的处理。

5、若是解析出的是链接（URL）,则把URL交给Scheduler等待抓取

**2. 安装Scrapy**

使用以下命令:

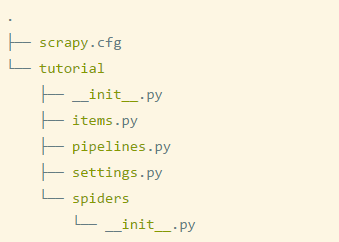


**3. Scrapy Tutorial**

在抓取之前, 你需要新建一个Scrapy工程. 进入一个你想用来保存代码的目录，然后执行：

$ scrapy startproject tutorial

这个命令会在当前目录下创建一个新目录 tutorial, 它的结构如下:



这些文件主要是：

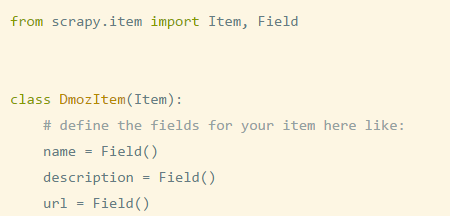
* scrapy.cfg: 项目配置文件
* tutorial/: 项目python模块, 之后您将在此加入代码
* tutorial/items.py: 项目items文件
* tutorial/pipelines.py: 项目管道文件
* tutorial/settings.py: 项目配置文件
* tutorial/spiders: 放置spider的目录

**3.1. 定义Item**

Items是将要装载抓取的数据的容器，它工作方式像 python 里面的字典，但它提供更多的保护，比如对未定义的字段填充以防止拼写错误

通过创建scrapy.Item类, 并且定义类型为 scrapy.Field 的类属性来声明一个Item.

我们通过将需要的item模型化，来控制从 dmoz.org 获得的站点数据，比如我们要获得站点的名字，url 和网站描述，我们定义这三种属性的域。在 tutorial 目录下的 items.py 文件编辑



**3.2. 编写Spider**

Spider 是用户编写的类, 用于从一个域（或域组）中抓取信息, 定义了用于下载的URL的初步列表, 如何跟踪链接，以及如何来解析这些网页的内容用于提取items。

要建立一个 Spider，继承 scrapy.Spider 基类，并确定三个主要的、强制的属性：

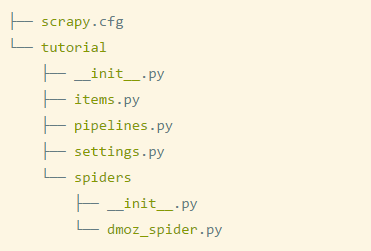
* name：爬虫的识别名，它必须是唯一的，在不同的爬虫中你必须定义不同的名字.
* start\_urls：包含了Spider在启动时进行爬取的url列表。因此，第一个被获取到的页面将是其中之一。后续的URL则从初始的URL获取到的数据中提取。我们可以利用正则表达式定义和过滤需要进行跟进的链接。
* parse()：是spider的一个方法。被调用时，每个初始URL完成下载后生成的 Response 对象将会作为唯一的参数传递给该函数。该方法负责解析返回的数据(response data)，提取数据(生成item)以及生成需要进一步处理的URL的 Request 对象。（这个方法负责解析返回的数据、匹配抓取的数据(解析为 item )并跟踪更多的 URL。）

在 /tutorial/tutorial/spiders 目录下创建 dmoz\_spider.py



**3.3. 爬取**

当前项目结构



到项目根目录, 然后运行命令:

$ scrapy crawl dmoz

运行结果:

