爬虫：从网上下载并提取项目感兴趣的海量数据

开发爬虫的方法很多：比如使用python自带的urllib和re；专业的爬虫开发框架(Scrapy)整合了大量工具包（比如Twisted，lxml，cssselect等）。

既然有跑车，为何还要步行去远方。

步骤：

step1发送url请求，获取相应

step2使用某种技术（如正则表达式，XPath）来提取页面中感兴趣的信息

step3识别响应中的链接信息，顺着链接递归执行step1-step3

step4 使用多线程有效管理网络通信交互

re识别响应页面的链接信息不如XPath，XPath的相关知识参见《疯狂XML讲义》

# 了解Scrapy

pip install scrapy

pyOpenSSL-22.0.0 python用于支持ssl（security socket layer）的包

cryptography: python用于加密的库

lxml-4.8.0 一个处理xml，html文档的库，比python内置的xml模块好用

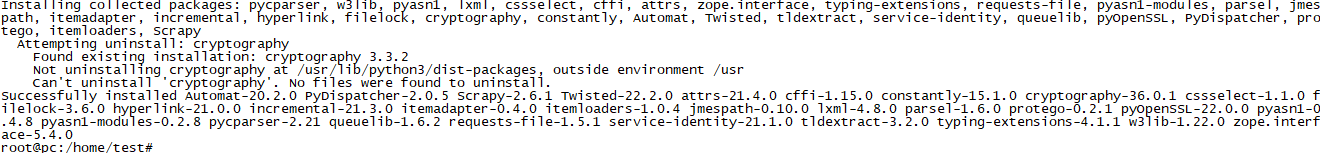
cssselect: python用于处理css选择器的扩展包

Twisted-22.2.0 为python提供基于事件驱动的网络引擎包

zope.interface-5.4.0 ：为python缺少接口而提供扩展的库

CFFI：python用于调用c的接口库





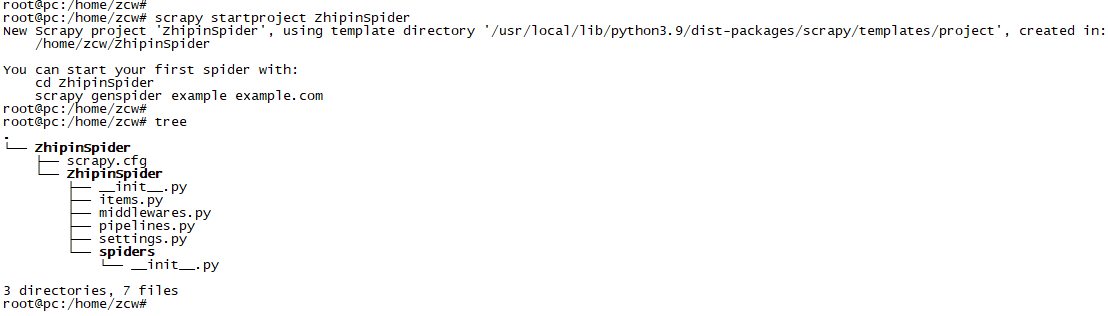
python3 -m pydoc -p 8899

http://localhost:8899/



# 使用爬虫爬取分和分析招聘信息

scrapy是命令，startproject是子命令，用于创建项目



scrapy.cfg: 项目总配置文件，通常无需修改

ZhipinSpider: 项目的python模块，程序将此处导入python代码

ZhipinSpider/settings.py: 项目配置文件，在该文减重进行项目相关的配置。

ZhipinSpider/items.py: 用于定义项目用到的Item类。Item类就是一个DTO（数据传输对象），通常就是定义N个属性，该类需要由开发者来定义。

ZhipinSpider/pipelines.py: 项目的管道文件，它负责处理爬取到的信息（用户编写）

ZhipinSpider/spiders: 在该目录下存放项目所需的蜘蛛 —— 蜘蛛负责抓取项目感兴趣的信息。

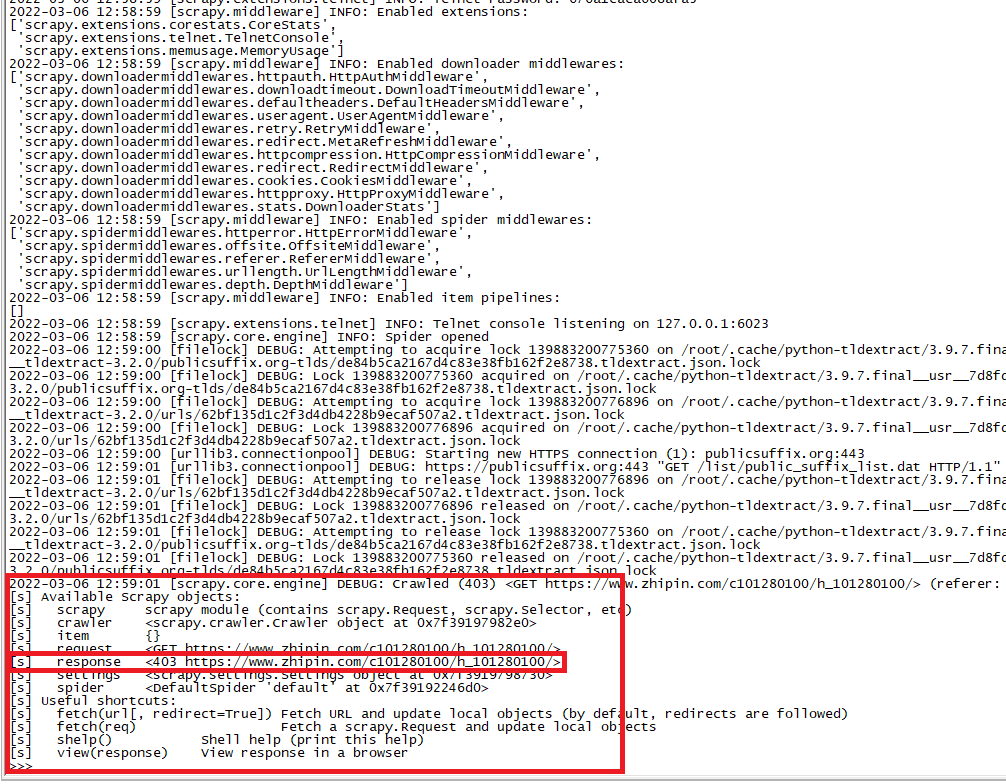
调用中间件 –>下载中间件 -> 蜘蛛中间件 ----Item----> pipeline

蜘蛛：负责从下载数据中提取有效信息，提取到的信息会由Scrapy引擎以Item对象的形式转交给Pipeline。

Pipeline：收到Item对象(包含蜘蛛提取的信息)后，将信息写入文件或数据库。

# 使用shell抓取网页

scrapy shell https://www.zhipin.com/c101280100/h\_101280100/



403说明目标网站开启了防爬虫

为了让Scrapy伪装成浏览器，需要在发送请求时设置User-Agent头，将User-Agent头的值设置为真实浏览器发送请求的User-Agent。

https://www.runoob.com/xpath/xpath-tutorial.html

XPath最实用的简化写法



XPath使用路径表达式在xml文档中进行导航

除了XPath，Scrapy也支持使用CSS选择器来匹配节点，但XPath匹配能力更强。

response.xpath('//div[@class="job-primary"]/div/h3/a/div/test()').extract()

response.css('div.job-primary>div.info-primary>h3.name span').extract()

# 本章小结

Scrapy是python领域专业的爬虫开发框架，scrapy框架已经完成爬虫的大部分通用工作，因此使用Scrapy开发爬虫项目简单方便。

使用Scrapy开发爬虫的核心步骤：

(1) 定义Item类

(2) 开发Spider，从网页上提取数据，也提取翻页连接

(3) 开发Pipeline，负责将提取的数据写入文件或数据库。

要区别对待静态网页和动态网页，两种网页的爬取方式略有差异。