**Overview**

Web crawling is a common technique to efficiently collect information from across the web. Here I wrote down some study notes while I was learning collecting data with Scrapy. An open source and collaborative framework for extracting the data.

**Installation**

***Install Scrapy on Windows***

I installed scrapy on Windows 8. Here are the steps:

1. Install Python 2.7.

Download link is [here](https://www.python.org/ftp/python/2.7/python-2.7.amd64.msi)

1. Add C:\python27\ and C:\python27/scripts folder to the system path:

From your computer: Control Panel > System > Advanced System Settings link(Advanced Tab) > Environmental Variables

1. Install OpenSSL:

* Download Win32 OpenSSL from [here](https://www.python.org/ftp/python/2.7/python-2.7.amd64.msi).

Please make sure download the right version for your Windows.

* Download Visual C++ 2008 redistributables for your version of Windows
* Download OpenSSL for your version of Windows
* Add the c:\openssl-win32\bin to your system Path.

1. Install Setuptools

* get-pip.py
* if get-pip.py not work, download ez\_setup.py from [here](https://bitbucket.org/pypa/setuptools/raw/bootstrap/ez_setup.py).Save it in C:\python27\scripts. Run C:\python27\scripts > python ez\_setup.py

1. Install PIP

* Download get-pip.py from [here](https://raw.github.com/pypa/pip/master/contrib/get-pip.py). Save it in C:\python27\scripts
* Run C:\python27\scripts > python get-pip.py

1. Install LXML

* Download LXML 3.5.0 from [here](http://www.lfd.uci.edu/~gohlke/pythonlibs/#lxml) for your version of windows.
* Run the installer

1. Install Twisted

* Download Twisted 14.0.2 from [here](http://twistedmatrix.com/Releases/Twisted/14.0/Twisted-14.0.2.win-amd64-py2.7.msi).

1. Install Scrapy:

* command: pip install scrapy

1. Install Mongo DB

* Please follow the following link to install mongoDB on your version of windows/linux from [here](https://docs.mongodb.org/manual/installation/).
* Download Robomongo, the cross-platform MongoDB manager from [here](https://robomongo.org/).

**Create Scrapy Project**

1. To create a scrapy project: *scrapy startproject {ProjectName}*
2. Defining Items: Specify data from items.py file within the project folder. Example:

import scrapy

from scrapy.item import Item, Field

class StackOverflowItem(scrapy.Item):

title = Field()

url =Field()

1. To crate spider: *scrapy genspider mydomain mydomain.com*
2. To locate element on the web.

Normally we locate the element from the web by Xpath or CSS selector. From Firefox, you can locate xpath directly by locate the element, right click, and select ‘View Xpath’. Or you can check the find the xpath from the browser console when select ‘Inspect Elements’. More details about xpath from [here](http://www.w3schools.com/xsl/xpath_intro.asp).

* Trying selectors in the Shell.

Scrapy shell is the built-in tool we can use to illustrate the use of the selectors.In order to use that, you have to install [IPython](http://ipython.org/) and [Scrapy shell](http://doc.scrapy.org/en/latest/topics/shell.html#topics-shell).

* To start a shell, you must go to the project’s top level directory and run:

scrapy shell “<http://www.testdomain.com>”

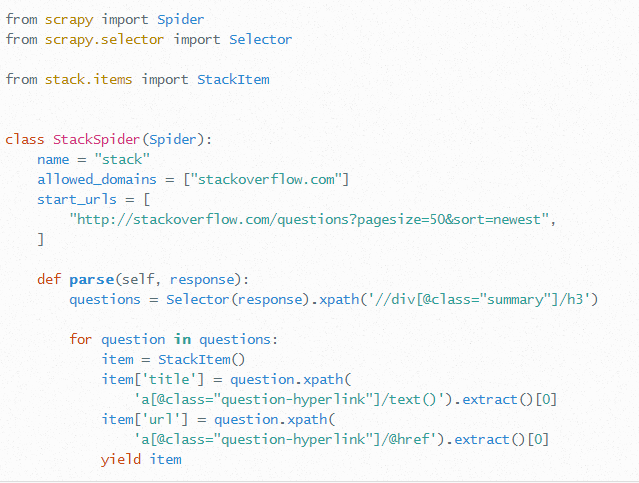
* To get response from target xpath:some examples:

response.xpath(‘//ul/li[@class=”someClass”]’)

response.xpath(‘//ul/li[@class=”someClass”]/text()’).extract()

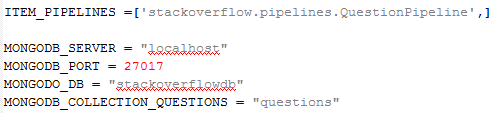
response.xpath(‘//ul/li[@class=”someClass”]/@href’).extract()

1. Sample of a spider

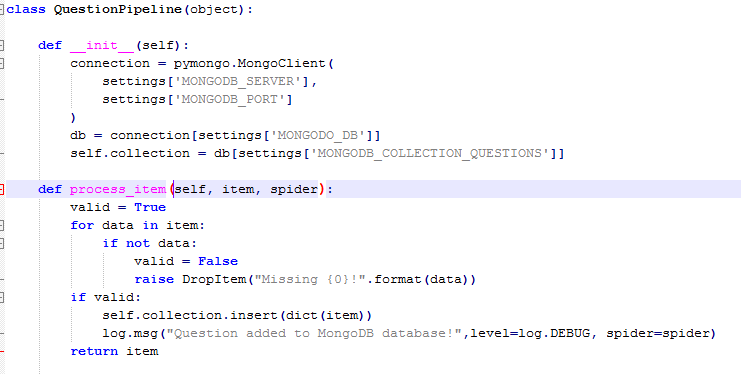


**Test and Store Data:**

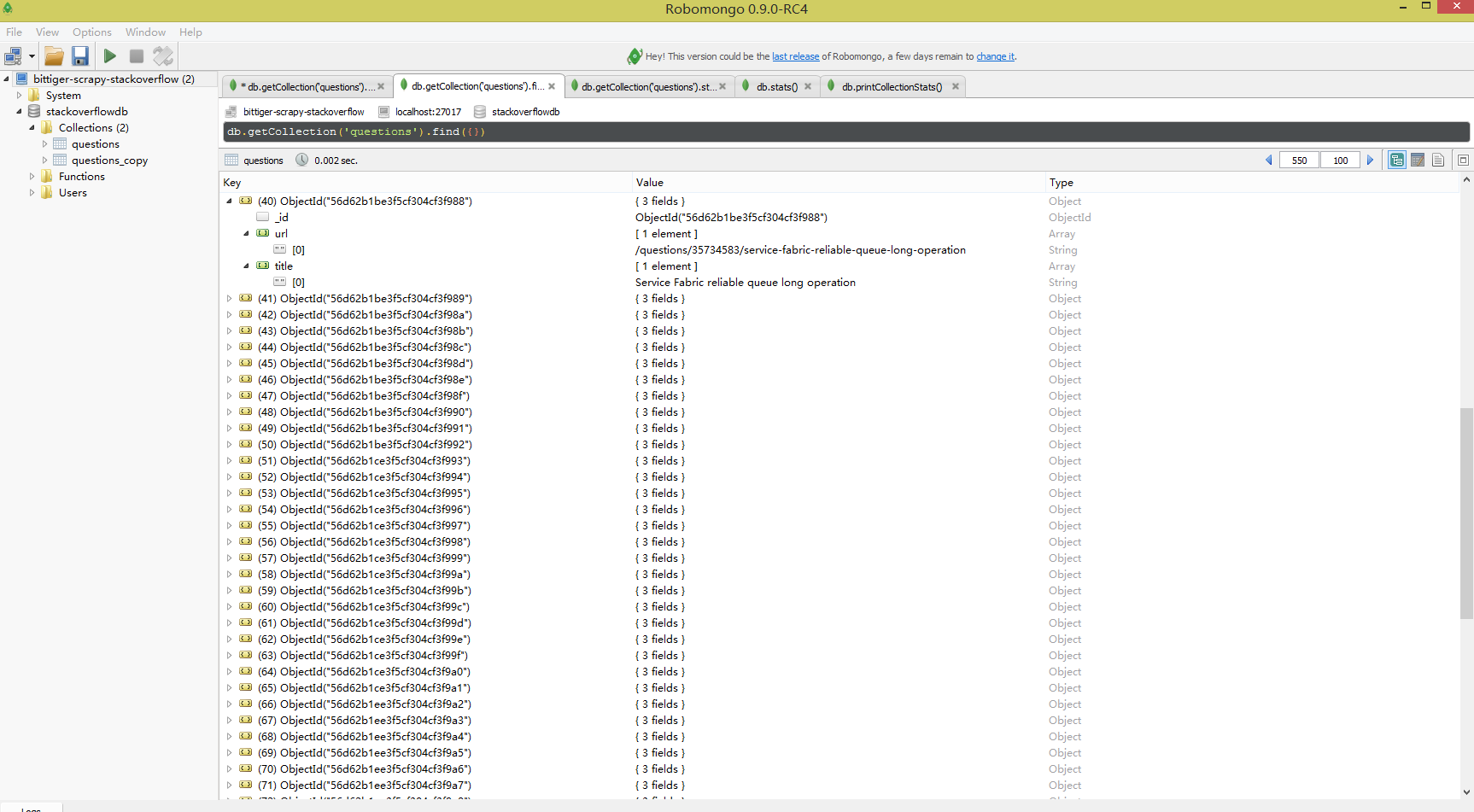
1. To run the test: run the command within your project directory: *scrapy crawl {SpiderName}*
2. To store data into a JSON file: scrapy crawl {SpiderName} -o items.json -t json
3. To store data in MongoDB:
   1. specify the pipeline and add the database settings in the settings.py:



1. Pipeline Management: This is the connection between the spider and the database settings. One sample:



1. Run the test: *scrapy crawl {spiderName}*
2. View data in database: Open Robomongo, connect to the server. And View the results:



**Crawl dynamic web with Splash:**

Splash is a javascript rendering service that can be used for crawling dynamic web which involved with javascript.

***Splash installation***:

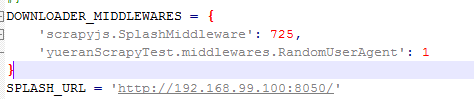
For OS X+Docker

1. Install [Docker](https://www.docker.com/).(via [Boot2Docker)](http://boot2docker.io/)
2. Pull the splash image from docker: *docker pull scrapinghub/splash*
3. Start the container: *docker run -p 5023:5023 -p 8051:8051 scrapinghub/splash*
4. Figure out the ip address of boot2docker: docker-machine ip

For more detail about how to install splash, please check [here](http://splash.readthedocs.org/en/latest/install.html).

***ScrapyJS installation***:

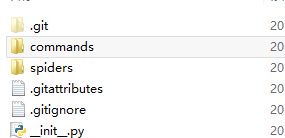
1. Install ScrapyJS: pip install scrapyjs
2. Modify Scrapy settings.py:



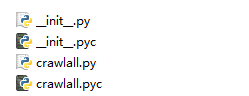
For more detail about splash, please check [here](http://splash.readthedocs.org/en/latest/index.html).

**Add custom commands:**

1. Create a ‘command’ folder with the same level of the ‘spiders’ directory



1. Create \_\_init\_\_.py under ‘commands’ directory:
2. Create a command\_name.py under ‘command’ directory



1. Add COMMANDS\_MODULE into the settings.py file



**Crawling multiple spiders at the same time**:

There are many ways to do that.

1. Run Scrapy from a script
2. Running multiple spiders in the same process
3. Running multiple spiders with custom command.

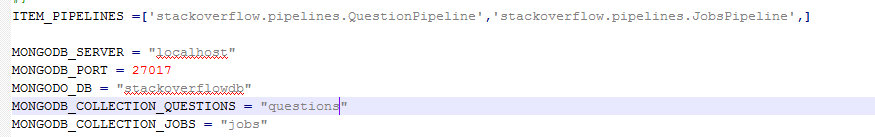
I used the third one in my project.

Please refer to <http://www.cnblogs.com/rwxwsblog/p/4578764.html> for more detail.

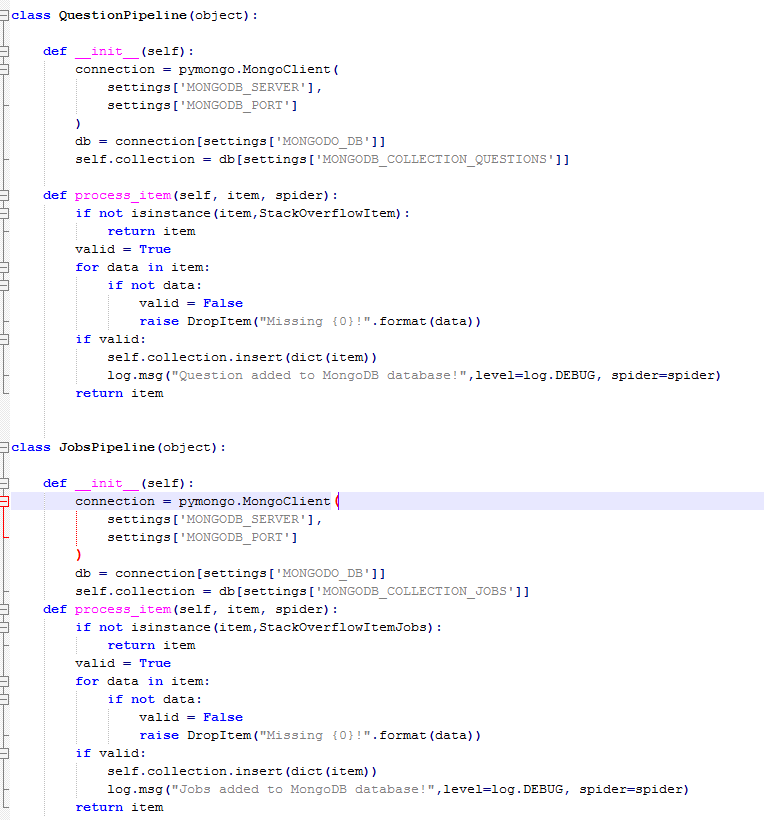
**Use different collections in mongodb**

Sometimes, you may need to store data in different collections based on item type. In order to do that, you can:

1. Configure the mongodb collections in settings.py page:



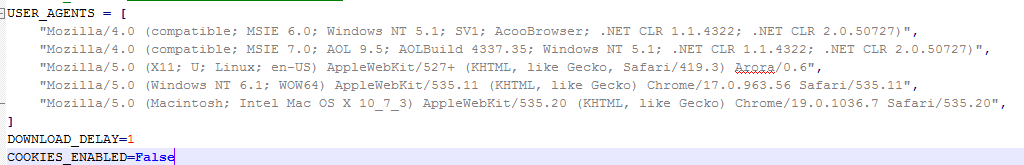
1. Configure pipelines.py



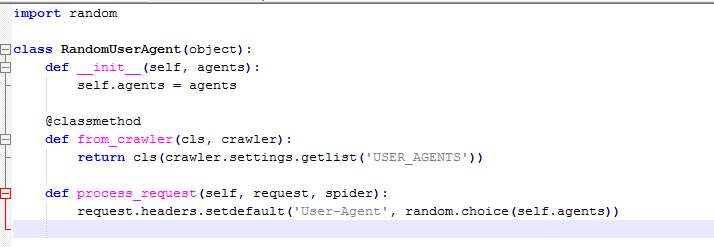
**Proxy**

Sometimes you may need to setup Proxy so that your ip would be less likely to be blocked. Here is how:

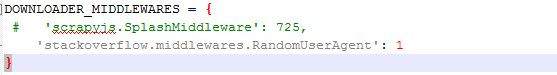
1. From settings.py page:
   1. Setup USER\_AGENTS
   2. Setup DOWNLOAD\_DELAY
   3. Setup COOKIES\_ENABLED to false



1. Create a middleware.py file with the same level of the ‘spiders’ directory
2. Add the code below in the middleware.py file:



1. Add middleware settings in settings.py:



**Other**

1. To remove /r/u?

Use map(Unicode.strip,…), for example:



**Reference:**

1. <http://venkateshtv.com/?p=43>
2. <http://doc.scrapy.org/en/latest/>
3. <https://realpython.com/blog/python/web-scraping-with-scrapy-and-mongodb/>
4. <http://splash.readthedocs.org/en/latest/index.html>
5. <http://www.cnblogs.com/rwxwsblog/p/4578764.html>
6. <http://blog.csdn.net/iefreer/article/details/20677943>
7. <http://stackoverflow.com/questions/19653963/use-different-collections-in-mongodb-depending-on-item-type>