INSTRUCTION DOCUMENT

Scrapy

Contents

[I Cài đặt Scrapy 3](#_Toc49266524)

[1 Window 3](#_Toc49266525)

[2 Ubuntu 3](#_Toc49266526)

[3 Kiểm tra cài đặt 4](#_Toc49266527)

[II Hướng dẫn tạo project 4](#_Toc49266529)

[1 Tạo project Scrapy 5](#_Toc49266530)

[2 Tạo spider 5](#_Toc49266531)

[3 Chạy spider 7](#_Toc49266532)

[III Scrapy shell 9](#_Toc49266533)

[IV Item Pipeline 10](#_Toc49266534)

[1 Item 10](#_Toc49266535)

[2 ItemLoader 11](#_Toc49266536)

[3 Item Pipeline 11](#_Toc49266537)

[V Settings 12](#_Toc49266538)

# Cài đặt Scrapy

Yêu cầu Python 3.5.2+

## Window

Cài đặt python có thể vào <https://www.python.org/> để cài đặt python trên thiết bị

Để cài đặt Scrapy ta có 2 cách:

Cách 1: sử dụng thư viện pip của python (có sẵn sau khi cài python), mở Command line:

Gõ lệnh: pip install Scrapy

Cách 2 nếu bạn sử dụng [Anaconda](https://docs.anaconda.com/anaconda/) or [Miniconda](https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html), thì có thể cài gói conda-forge để cài Scrapy, mở Command line:

Gõ lệnh: conda install -c conda-forge scrapy

## Ubuntu

Cài đặt python trên ubuntu đơn giản mở terminal và gõ lệnh: sudo apt-get install Python3

Nếu bạn sử dụng gói pip của python thì có thể cài thêm gói pip, mở terminal:

Gõ lệnh: sudo apt install python3-pip

Để cài đặt Scrapy bằng thư viện pip, tiếp tục:

Gõ lệnh: pip3 install Scrapy

## Kiểm tra cài đặt

Để kiểm tra đã cài đặt python thành công gõ lệnh: python –version

Nếu nó ra hiển thị phiên bản của python thì cài đặt thành công

Để kiểm tra đã cài đặt scrapy thành công gõ lệnh: scrapy

Nếu nó hiển thị các lệnh như Figure 1 thì bạn đã cài đặt thành công

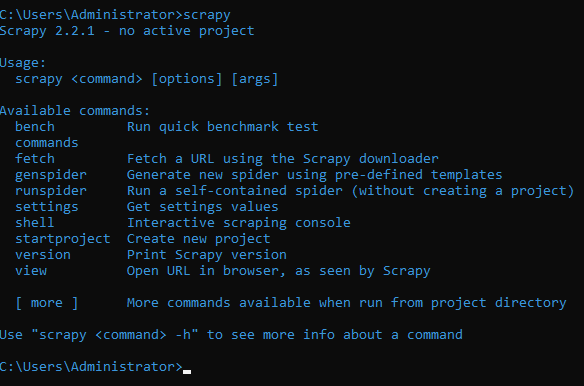


Figure 1 Scrapy version



# Hướng dẫn tạo project

Mở Command Line nếu bạn sử dụng Window, hoặc Terminal nếu bạn sử dụng Ubuntu

## Tạo project Scrapy

Để tạo một project Scrapy mới sử dụng câu lệnh:

scrapy startproject demo

Nó sẽ tự động tạo project tên demo có cấu trúc thư mục như sau:

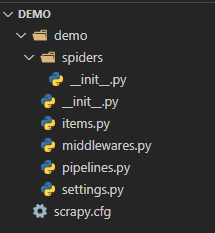


Figure 2 Cây thư mục

* File *scrapy.org* là configure của thư mục
* File items.py chứa các class item được cào về
* File middlewares.py chứa các middlewares để xử lý các extension
* File pipelines.py chưa các class pipeline cho việc xử lý các item sau khi cào về
* File settings.py cho phép configure các biến toàn cục mà cả project phụ thuộc vào
* File \_\_init\_\_.py là file khởi tạo
* Thư mục spiders chưa các spider cho việc thực hiện các request để cào dữ liệu về. Đây là các file làm việc hoạt động chính.

## Tạo spider

**Spider**: Là các class spider dùng để phân tích và trích xuất các thành phần mong muốn.

Để tạo mộ spider mới sử dụng câu lệnh:

scrapy genspider quotes quotes.toscrape.com

Nó sẽ tự động tạo ra một file *quotes.py* nằm trong thư mục spiders như *Figure 3* trong đó quotes là tên spider và quotes.toscrape.com là URL cần cào dữ liệu về.

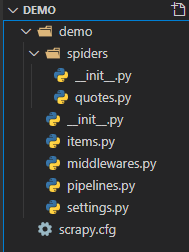


Figure 4 New spider class quotes

Nội dung bên trong file quotes.py

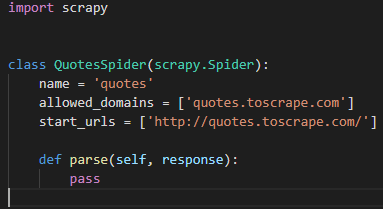


Figure 5 File quotes.py

* *name* là tên spider, thường được dùng để thực hiện cào dữ liệu.
* *allowed\_domains* là tên miền chính của source muốn cào về.
* *start\_urls* là url mà gửi request đầu tiên để bắt đầu cào, có thể có nhiều start\_url.
* hàm *parse* là hàm callback thường được gọi để xử lý các response sau khi được nhận từ mỗi request được gửi bởi spider.

Ngoài ra có thể sử dụng lệnh scrapy list để hiển thị danh sách các spider đã tạo.

## Chạy spider

Để spider thực hiện việc cào dữ liệu sử dụng câu lệnh

scrapy crawl quotes

Trong đó ‘quotes’ là tên spider thực hiện việc cào dữ liệu crawl là lệnh thực hiện việc cào dữ liệu.

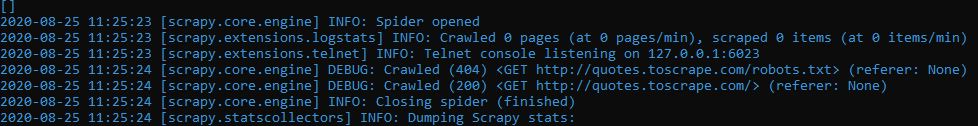


Figure 6 Request tới site

Thử thực hiện việc lấy dữ liệu từ trang website, ở đây thử lấy dữ liệu trên trang <http://quotes.toscrape.com/> đã được khỏi tạo cùng với spider quotes ở trên.

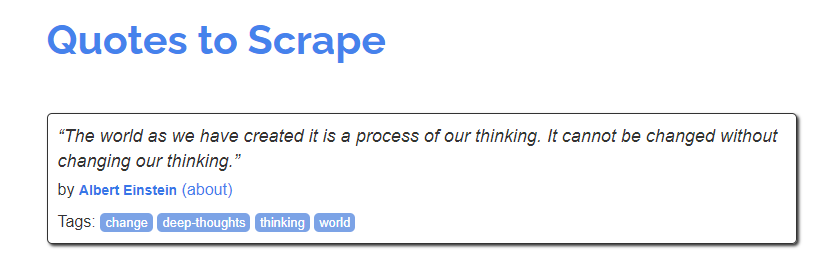


Figure 7 <http://quotes.toscrape.com/>

Thử gõ đoạn code sau vào hàm *parse* của spider *quote*s:

    def parse(self, response):

        quote = response.xpath('//\*[@class="text"]/text()').extract\_first()

        author = response.xpath('//\*[@class="author"]/text()').extract\_first()

        yield{

            'quote': quote,

            'author': author

        }

Sau đó gõ lại lệnh scrapy crawl quotes để run lại spider.

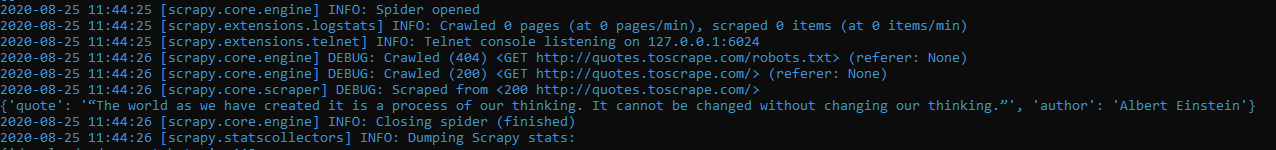


Figure 8 Crawl quote

Thấy được dòng dữ liệu cần cào về như Figure 9 , như vậy là cào được dữ liệu về thành công.

Nếu muốn lưu lại dữ liệu được cào về, có thể dùng **Feed Export** là cách đơn giản đễ lưu trữ, sử dụng bằng cách thêm tham biến –o <file-name>.<format> ở câu lệnh crawl.

Ví dụ:

* muốn lưu dữ liệu về dạng csv: scrapy crawl quotes –o item.csv
* muốn lưu dữ liệu về dạng json: scrapy crawl quotes –o item.json

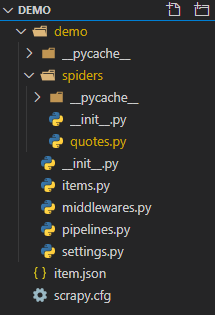
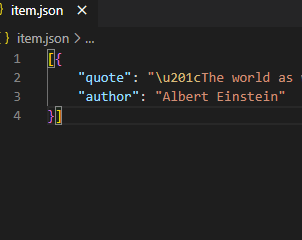
 

Figure 10 item.json

# Scrapy shell

Bên cạnh việc khỏi tạo một dự án scrapy, còn có thể sử dụng scrapy shell để thực hiện việc cào dữ liệu trên Command Line hoặc Terminal (có thể dùng để test).

**Khởi tạo:** Để khỏi tạo scrapy shell, gõ lệnh: scrapy shell

Sau khi khỏi tạo scrapy shell thành công, gõ lệnh: fetch(“<url>”) để thực hiện việc request tới site muốn cào dữ liệu.

Ví dụ fetch tới site <http://quotes.toscrape.com/>

Có thể gõ response để kiểm tra kết nối, nếu status là 200 thì kết nối thành công

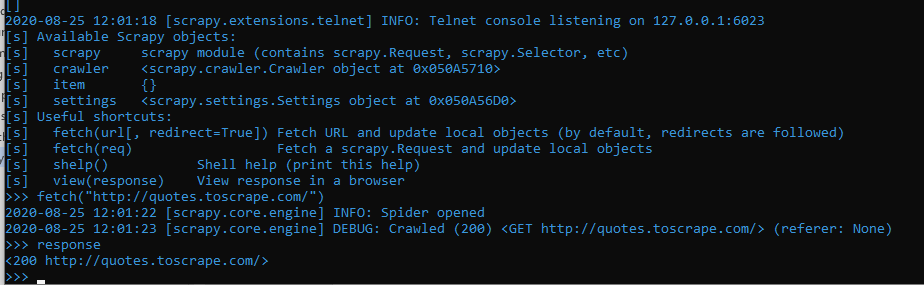


Figure 11 scrapy shell

**Crawl:** Để lấy dữ liệu thử gõ lệnh: response.xpath(‘//\*[@class=”text”]/text()’).extract() để các quotes trên site.

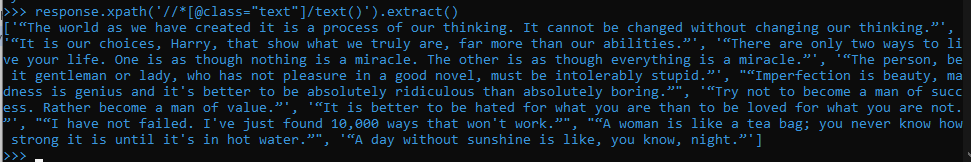


Figure 12 crawl quotes on shell

Giải thích câu lệnh:

* response là tham số được trả về sau khi gửi request từ spider.
* xpath là một cơ chế để chịn lấy thành phần từ response (F12 để xem inspect của site). Ngoài ra còn có nhiều cơ chế khác như lấy theo css.
* ‘//\*[@class=”text”]/text()’ là một **xpath** thực hiện tìm tất cả selector có tag class với value là “text” sau đó lấy text() nội tại của tag đó.
* extract() là phương thức thực hiện việc rút trích ra các phần tử đó từ selector. Như ví dụ trên thì sẽ lấy ra danh sách các quotes (extract\_first() để chỉ rút trích phần tử đầu tiên ).

Lưu ý: có thể tham khảo thêm về xpath để hiểu hơn về cách sử dụng.

**Exit:** Để thoát khỏi scrapy shell, dùng câu lệnh exit() để kết thúc.

# Item Pipeline

Sau khi spider cào dữ liệu, dữ liệu sẽ được gửi tới các item pipeline. Item Pipeline xử lý các item thông qua nhiều phần và được thực thi tuần tự.

Mỗi Item Pipeline là một class thực hiện các phương thức đơn giản, nó thực hiện một số hành động lên item và có thể quyết định xem item đó được đi tiếp hoặc là drop.

Để gửi các dữ liệu được cào về từ spider tới pipeline phải gửi thông qua ItemLoader bằng đối tượng Item

## Item

Item là object được xác định bằng 1 cặp key-value, có thể định nghĩa type bất kì cho item. Có thể định nghĩa 1 item trong file items.py

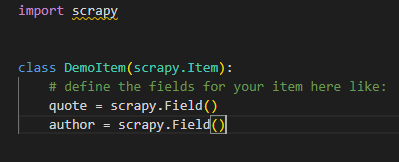


Figure 13items.py

Key: quote, author ; Value: Field

## ItemLoader

ItemLoader cung cấp một cơ chế thuận tiện cho việc lưu trữ item.

Để sử dụng ItemLoader, phải import ItemLoader từ scrapy.loader

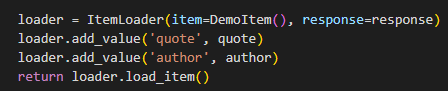


Figure 14 ItemLoader

Giải thích:

* loader = ItemLoader(item=DemoItem(), response=response thực hiện việc khỏi tạo 1 ItemLoader với kiểu dữ liệu là DemoItem
* loader.add\_value('quote', quote) thêm dữ liệu vào key ‘quote’ bằng phương thức add\_value()
* loader.load\_item() trả về với dữ liệu đã được lưu trữ trước đó bỏi phương thức add\_value()

## Item Pipeline

Mỗi Item Pipeline là một class thực hiện các phương thức đơn giản, nó thực hiện một số hành động lên item và có thể quyết định xem item đó được đi tiếp hoặc là drop.

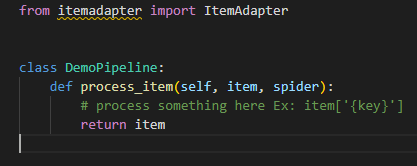


Figure 15 pipelines.py

Để spider có thể gửi Item đến Item Pipeline, bạn phải configure lại biến môi trường trong file settings.py

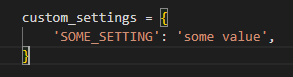


Figure 16 settings.py

Giải thích:

* ‘demo.pipelines.DemoPipeline’ là import class pipeline mà bạn muốn dùng
* 300 là thứ tự thực hiện từ giá trị thấp đến cao trong khoảng 0-1000
* Có thể có 1 hoặc nhiều item pipeline.

Sau khi configure trong settings.py chỉ cần chạy spider là có thể thực hiện cào dữ liều và gửi item tới item pipeline.

# Settings

Scrapy Settings cho phép hiệu chỉnh lại hành vi của tất cả các thành phần của Scrapy, bao gồm cả core, extensions, pipelines và bản thân spider. Cơ sở hạ tầng cung cấp những biến toàn cục dạng key value để configuration cho các đoạn code.

Bên cạnh việc configure pipeline như đã đề cập ở phần IV, còn có thể configure thêm nhiều thứ khác.

* ROBOTSTXT\_OBEY = True : biến này configure để xem spider của mình có tuân thủ theo file robots.txt (file có chức năng quản lý lưu lượng truy cập) của site muốn cào hay không (mặc định là True).
* DOWNLOAD\_DELAY = 3 : thời gian chờ giữa các lần dowload từ site (mặc định là 3s)
* COOKIES\_ENABLED = False : tùy chỉnh cookie (mặc định là False)
* USER\_AGENT = 'demo (+http://www.yourdomain.com)' : là là phần mềm hoạt động thay mặt cho người dùng ( có thể dùng user\_agent này ”[**Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.135 Safari/537.36**](https://developers.whatismybrowser.com/useragents/parse/?analyse-my-user-agent=yes)“)

Lưu ý: nếu configure trong file setttings.py thì tất cả các spider đều thực hiện theo các điều kiện đó. Trong trường hợp bạn muốn configure riêng cho từng spider thì ban thêm một thuộc tính là custom\_settings và truyền vào cặp key-value để custom theo ý muốn.

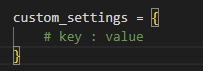


Figure 17 custom\_settings trong class spider