**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**BÁO CÁO PROJECT II**

**Tìm hiểu thư viện Scrapy trong việc cào dữ liệu web**

**PHẠM MINH KHÔI – 20183566 – IT1-02-K63**

Khoi.pm183566@sis.hust.edu.vn

**Ngành Khoa học Máy tính**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | TS. Phạm Quang Dũng |
| **Bộ môn :** | Project 2 |
| **Viện :** | Công nghệ thông tin và Truyền thông |

**HÀ NỘI, 05/2021**

**Lời cảm ơn**

Trong thời gian làm bài tập lớn môn Project II, em đã nhận được rất nhiều sự giúp đỡ, đóng góp ý kiến của thầy và các bạn.

Em xin gửi lời cảm ơn chân thành đến TS. Phạm Quang Dũng, giảng viên Viện Công nghệ thông tin và truyền thông, trường Đại học Bách khoa Hà Nội, người trực tiếp hướng dẫn em hoàn thành bài tập lớn môn Project II.

Em cũng xin cảm ơn các thầy cô, bạn bè đã luôn tạo điều kiện quan tâm, giúp đỡ để em hoàn thành học phần này.

Em rất vinh dự nếu nhận được sự góp ý của thầy cô và các bạn để hoàn thiện đề tài này hơn.

Em xin chân thành cảm ơn!

**Tóm tắt nội dung**

* Vấn đề cần thực hiện: nghiên cứu và tìm hiểu thư viện Scrapy và thực hiện cào lấy dữ liệu từ một số trang web.
* Phương pháp thực hiện:
  + Tìm hiểu tổng quan về các thư viện python có sẵn: beautiful soup, selenium, scrapy.
  + Cài đặt thử nghiệm chương trình cào dữ liệu bằng ngôn ngữ python.
* Kết quả: Xuất ra file json chứa dữ liệu lấy được.
* Tính thực tế: Chương trình giúp trong việc huấn luyện mô hình AI, Deeplearning bằng việc cung cấp tập dữ liệu.
* Định hướng phát triển: Tìm cách cải tiến mới tối ưu tốc độ và bộ nhớ, đồng thời kết hợp với các phương pháp khác trong việc lấy dữ liệu từ các trang web giới hạn…

Sinh viên thực hiện

Ký và ghi rõ họ tên

Phạm Minh Khôi

**MỤC LỤC**

[CHƯƠNG 1. PHÁT BIỂU VÀ GIẢI QUYẾT BÀI TOÁN 4](#_Toc60267668)

[1.1 Phát biểu bài toán 4](#_Toc60267669)

[1.2 Phương pháp thực hiện 4](#_Toc60267670)

[1.2.1 Tạo project scrapy 4](#_Toc60267671)

[1.2.2 Tạo spider 4](#_Toc60267672)

[1.2.3 Xác định data để lấy về 5](#_Toc60267673)

[1.2.4 Tiến hành cào dữ liệu 6](#_Toc60267673)

[CHƯƠNG 2. KẾT QUẢ THỬ NGHIỆM VÀ ĐÁNH GIÁ 7](#_Toc60267674)

[CHƯƠNG 3. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 8](#_Toc60267674)

[TÀI LIỆU THAM KHẢO 8](#_Toc60267677)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Khởi tạo spider. 5](#_Toc60267678)

[Hình 2. Items.py 5](#_Toc60267679)

[Hình 3&4. File TGDD\_spider.py 6](#_Toc60267680)

[Hình 5. Kết quả file data.json 7](#_Toc60267681)

[Hình 6. File settings.py. 7](#_Toc60267682)

**Chương 1.Phát biểu vấn đề và giải quyết bài toán.**

**1.1 Phát biểu bài toán**

* Cào dữ liệu (Crawl Data) là một thuật ngữ không còn mới trong công nghệ nói chung và khoa học dữ liệu nói riêng. Vì crawl là kỹ thuật mà các con robots của các công cụ tìm kiếm sử dụng như: Google, Facebook, Amazon… Công việc chính của Crawl là thu thập dữ liệu và tiến hành phân tích mã nguồn Html nhằm lọc ra thông tin theo yêu cầu của Search Engine.
* Khó khăn trong việc crawl data: crawl data lấy từ dữ liệu có cấu trúc html nên khi website thay đổi cấu trúc thì chương trình crawl phải update theo cho thích hợp với thay đổi. Một số trang web lớn như FaceBook, Amazon, Tiki chặn crawl data từ trang web của họ (chủ yếu là tính tổng số request gửi đến server trên đơn vị thời gian) khiến cho địa chỉ proxi dễ bị ban.
* Khác biệt giữa Crawl và Scrape dữ liệu:
  + Web Crawling là quá trình thu thập thông tin từ các website theo các đường links cho trước. Các web crawler sẽ tải toàn bộ nội dung và tìm kiếm thêm các đường links bên trong để tiếp tục truy cập. Dữ liệu khi được tải về sẽ được đánh chỉ số rồi lưu vào cơ sở dữ liệu.
  + Web Scraping cũng thực hiện việc tìm kiếm và thu thập dữ liệu nhưng khác với crawling, nó không thu thập toàn bộ thông tin mà chỉ lấy những thứ thật cần thiết và phù hợp với mục đích sử dụng. Trong scraping chúng ta kết hợp với Data Extraction (trích xuất dữ liệu) để tập trung các nội dung cần thiết.
* Các công cụ phổ biến: Scrapy, BeautifulSoup, Selenium, Jsoup, ApacheNutch.

**1.2 Phương pháp thực hiện**

**1.2.1 Tạo project scrapy**

Cài đặt thư viện scrapy với câu lệnh : pip install scrapy

Tạo 1 project scrapy với tên TGDDtutorial: scrapy startproject TGDDtutorial

Project mà chúng ta vừa tạo ra có cấu trúc như sau:

demo\_scrapy/

scrapy.cfg # deploy configuration file

demo\_scrapy/ # project's Python module

\_\_init\_\_.py

items.py # project items definition file

middlewares.py # project middlewares file

pipelines.py # project pipelines file

settings.py # project settings file

spiders/ # a directory where you'll later put your spider

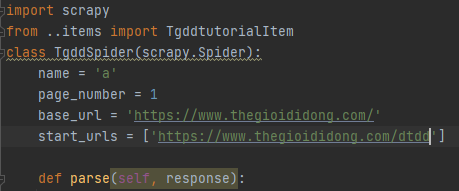
\_\_init\_\_.py

**1.2.2 Tạo spider**

Sau khi tạo được project, chúng ta tạo 1 spider để crawl dữ liệu bằng câu lệnh:

scrapy genspider a [www.thegioididong.com](http://www.thegioididong.com)/dtdd

Câu lệnh trên đã tạo ra một file python trong thư mục spiders như sau:

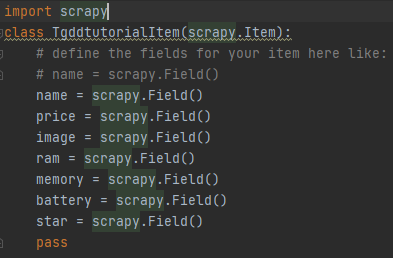


Hình 1. Khởi tạo spider.

* Spider có tên là “a” để chạy trên terminal.
* Start\_urls là địa chỉ bắt đầu cho spider, ở đây cụ thể là crawl điện thoại di động từ web thegioididong.
* Parse() là function chứa code điều khiển spider cào dữ liệu .

**1.2.3 Xác định data để lấy về**

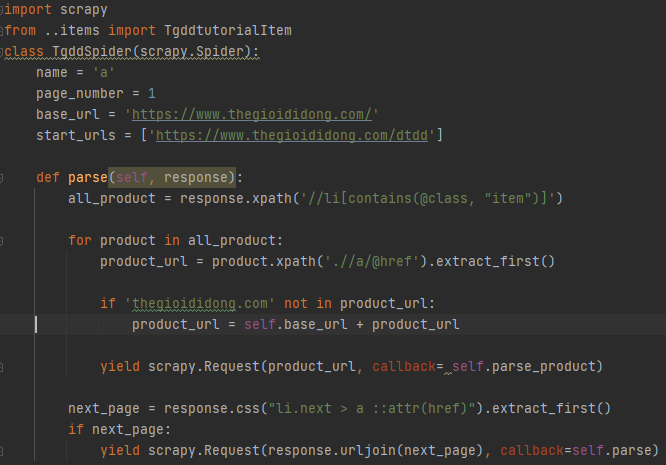
Tùy thuộc vào mục đích của người sử dụng, chúng ta sẽ lấy những data cần thiết về, sau khi đã xác định được thì chúng ta sẽ define trong file items.py:

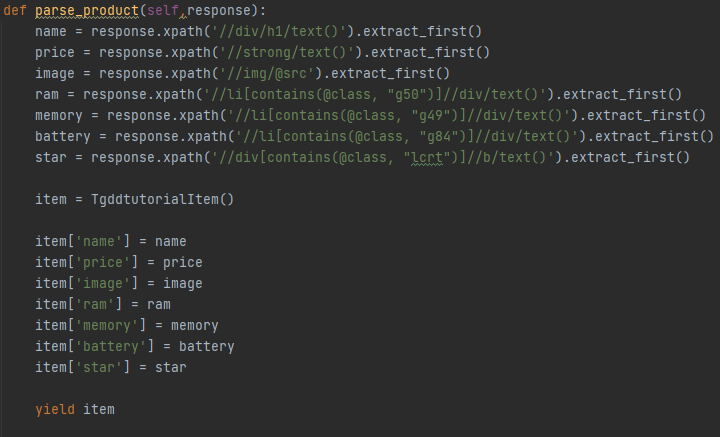


Hình 2. Items.py

**1.2.4 Tiến hành cào dữ liệu**

Trước khi tiến hành, chúng ta phải xác định được đối tượng đó trên DOM của web, bằng cách sử dụng css (hoặc xpath đều được). Sau khi tiến hành, chúng ta được file spiders/TGDD\_spider.py như sau:





Hình 3 & 4. File TGDD\_spider.py

* Parse(): Vì các dữ liệu của từng sản phẩm bên ngoài không chứa đủ dữ liệu cần thiết, nên chúng ta gửi scrapy.Request vào từng sản phẩm thông qua url của nó.
* Next\_page: để di chuyển tới các trang sản phẩm phía sau (thực hiện được với những trang được đánh chỉ số/ tổ chức theo hàng ngang, chưa chắc đã thực hiện được với những trang tổ chức theo hàng dọc).
* Parse\_product(): trong hàm này chứa code tiến hành crawl những dữ liệu đã define trong items.py.

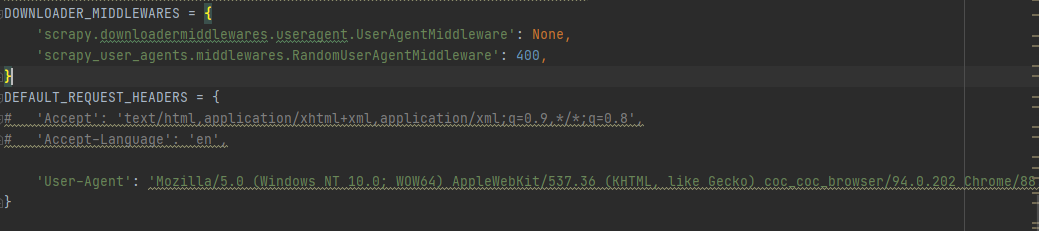
Để crawl, chúng ta chạy câu lệnh: scrapy crawl a –o data.json

Scrapy cũng hỗ trợ xuất dữ liệu sang các định dạng khác như csv hoặc xml.



Hình 5. Kết quả file data.json

Tuy nhiên trong quá trình crawl, một số trang web sẽ chặn việc cào dữ liệu, vì vậy chúng ta sẽ thêm một ít code nữa vào file settings.py:



Hình 6.File settings.py

Cụ thể:

* Downloader\_middlewares sẽ tạo ra các phiên bản giả lập các dòng máy như sony, samsung, window… để gửi request tới web.
* Default\_request\_headers sẽ cấu hình thêm User-agent có chức năng tương tự như trên.

**Chương 2. Kết quả thử nghiệm và đánh giá.**

Chương trình được chạy thử trên máy tính với CPU Intel Core(TM) i5-4300 2GHz, RAM 4GiB. Hệ điều hành Win10Pro, 64bit.

Hiện tại em đã crawl được 3 bản demo từ thế giới di động, báo dân trí và amazon. Tất cả đều chạy được thành công và không bị giới hạn.

Vẫn tồn tại một số trang web có thiết kế theo chiều dọc (tức là có phần xem thêm chứ không có số trang) thì em chỉ lấy được dữ liệu ở những trang đầu tiên, vì vậy vẫn cần phải cải thiện để có thể ứng dụng trong thực tế. Cũng có những trang web yêu cầu đăng nhập tài khoản để truy cập vào từng item, vì vậy cũng cần phải cung cấp thêm phương thức tấn công thông qua lớp đệm ấy (gửi request bằng cách tự động thay vì thủ công như tạo một tài khoản chỉ để dùng 1 lần). Tuy nhiên, em cũng đã cố gắng hoàn thành trong thời gian được giao và thu được kết quả như trên.

**Chương 3. Kết luận và hướng phát triển.**

* Về cơ bản đã xây dựng được một chương trình giải quyết được đầy đủ các yêu cầu của bài toán.
* Ưu điểm là cách thức hoạt động dễ nhìn, dễ hiểu; các thành phần chương trình đã được phân chia rõ ràng từng chức năng nhờ vào thư viện đã cài đặt sẵn; tốc độ cào nhanh, phù hợp với những yêu cầu cần bộ dữ liệu lớn.
* Nhược điểm là có độ tương thích không lớn với những trang web sử dụng javascript để render, dễ dẫn đến việc trả về một bộ data không có dữ liệu. Từ đó có thể phát triển thêm theo các hướng:
  + Sử dụng kèm với headless browser như Splash để chờ trang web render ra nội dung và cookie, rồi gửi đống html này về crawler để bóc tách như bình thường.
  + Sử dụng các ứng dụng thêm từ web <https://www.scraperapi.com/> hỗ trợ tới 5000 API calls, 10 concurrent request, proxies, Js rendering, …

Source code: <https://github.com/ghuioio/project2>

**TÀI LIỆU THAM KHẢO**

[1] <https://viblo.asia/p/su-ket-hop-hoan-hao-cua-scrapy-va-splash-giai-phap-toi-uu-voi-trang-web-su-dung-javascript-Qbq5Qa8w5D8n>

[2] <https://scrapy.org/>

[3] https://viblo.asia/p/tap-tanh-crawl-du-lieu-voi-scrapy-framework-bWrZnW7rlxw

[4] <https://dev.to/davidmm1707/python-scrapy-tutorial-for-beginners-01-creating-your-first-spider-1apd>

[5] <https://www.youtube.com/watch?v=ve_0h4Y8nuI&list=PLhTjy8cBISEqkN-5Ku_kXG4QW33sxQo0t>

[6] <https://dev.to/iankerins/how-to-scrape-amazon-at-scale-with-python-scrapy-and-never-get-banned-44cm>