**CRAWL DATA TRONG LARAVEL**

# Crawl data là gì ?

## Khái niệm

Crawl data (hay còn gọi là web scraping) là quá trình tự động thu thập dữ liệu từ các trang web thông qua việc truy cập và phân tích cấu trúc HTML của trang web đó. Kỹ thuật này cho phép bạn tự động lấy thông tin từ các trang web và lưu trữ dữ liệu thu được để sử dụng cho mục đích phân tích, hiển thị, hoặc xử lý dữ liệu khác.

## Các bước

Quá trình crawl data thường bao gồm các bước sau:

### Gửi yêu cầu HTTP: Bắt đầu bằng việc gửi yêu cầu HTTP (thường là yêu cầu GET) đến URL của trang web mà bạn muốn thu thập dữ liệu.

### Lấy nội dung trang web: Khi nhận được phản hồi từ yêu cầu HTTP, bạn sẽ nhận được nội dung HTML của trang web đó. Bạn có thể lấy nội dung này thông qua API HTTP hoặc các thư viện HTTP trong ngôn ngữ lập trình của bạn.

### Phân tích HTML: Sau khi có nội dung HTML, bạn sẽ sử dụng các công cụ phân tích HTML để trích xuất dữ liệu mà bạn quan tâm. Công cụ phân tích HTML giúp bạn xác định các phần tử HTML, thuộc tính và cấu trúc của trang web để trích xuất thông tin.

### Trích xuất dữ liệu: Dựa trên cấu trúc HTML và các quy tắc phân tích, bạn sẽ lựa chọn và trích xuất các mục dữ liệu cần thiết từ trang web. Điều này có thể bao gồm lấy văn bản, hình ảnh, liên kết, bảng dữ liệu, hoặc bất kỳ loại dữ liệu nào khác có sẵn trên trang web.

### Lưu trữ và xử lý dữ liệu: Sau khi trích xuất dữ liệu từ trang web, bạn có thể lưu trữ nó vào cơ sở dữ liệu, tệp tin hoặc xử lý dữ liệu trực tiếp trong ứng dụng của bạn. Quá trình này có thể bao gồm lọc, chuyển đổi định dạng hoặc phân tích thêm dữ liệu thu thập được.

# Cài đặt thư viện

Cài đặt thư viện **Goutte**

|  |
| --- |
| **composer require fabpot/goutte** |

# Tạo controller xử lý

## Lệnh tạo

|  |
| --- |
| php artisan make:controller **CrawlDataController** |

## Cấu hình app\Http\Controllers\**CrawlDataController.php**

Crawl data tại địa chỉ <https://cellphones.com.vn/mobile.html?mobile_os_filter=iphone-ios>

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use Goutte\Client;  class CrawlDataController extends Controller  {      public function run(){          $client = new Client();          $crawler = $client->request('GET', 'https://cellphones.com.vn/mobile.html?mobile\_os\_filter=iphone-ios');          // Trích xuất dữ liệu từ HTML          $data = $crawler->filter('.product-item')->each(function ($node) {              return [                  'name' => $node->filter('.product\_\_name')->text(),                  'price' => $node->filter('.product\_\_price--show')->text(),              ];          });           dd($data);      }  } |

## Giải thích các đoạn code như sau

### **$client = new Client();** Tạo một đối tượng $client từ lớp Client trong thư viện Goutte. Đối tượng này sẽ được sử dụng để thực hiện các yêu cầu HTTP và crawl dữ liệu từ trang web

### **$crawler = $client->request('GET', 'https://cellphones.com.vn/mobile.html?mobile\_os\_filter=iphone-ios'); :** Gửi request với phương thức GET tới địa chỉ URL như trên và lấy dữ liệu HTML của web đó và gán cho biến $crawler

### **$crawler->filter('.product-item')->each(function ($node)**

|  |
| --- |
|  |

Ta thấy mỗi sản phẩm được bao trong một class là **product-item**, ta dùng filler để lọc những phần tử có class là **product-item**, và vì có nhiều phần tử có class là **product-item** nên nó là 1 mảng, dùng each để lặp qua mảng có các phần tử là **product-item** và trích xuất dữ liệu từ chúng

$node chính là giá trị của mỗi phần tử khi lặp qua

### $node->filter('.product\_\_name')->text() :

Lọc những class là **product\_\_name** , text() lấy dữ liệu text từ phần tử đó

|  |
| --- |
|  |

### $node->filter('.product\_\_price--show')->text()

Lọc những class là **product\_\_price--show**, text() lấy dữ liệu text từ phần tử đó

|  |
| --- |
|  |

### dd($data);

In ra dữ liệu đã lọc ở trên

## Route web.php

routes\**web.php**

|  |
| --- |
| use App\Http\Controllers\CrawlDataController;  Route::get('/crawl-data',[CrawlDataController::class, 'run']); |

## Kết quả màn hình

Truy cập trình duyệt tại địa chỉ <http://127.0.0.1:8000/crawl-data>

Ta thấy đã có dữ liệu của 20 sản phẩm, thông tin bao gồm name(tên sản phẩm) và price(giá), vì lúc nảy ta chỉ lọc ra có 2 tiêu chí thôi

|  |
| --- |
|  |

## Lấy giá trị của các thuộc tính trong thẻ HTML

|  |
| --- |
|  |

Lấy thuộc tính src

|  |
| --- |
| 'image' => $node->filter('img.product\_\_img')->attr('src') |

Lấy thuộc tính alt

|  |
| --- |
| 'descriptionImage' =>$node->filter('img.product\_\_img')->attr('alt') |

# Một số phương thức phổ biến

## filter($selector): Lọc các phần tử HTML dựa trên một CSS selector. Trả về một đối tượng Crawler mới chứa các phần tử đã lọc.

|  |
| --- |
| $crawler->filter('.listProduct-row.ajax-filter>.proloop') |

## each($closure): Lặp qua từng phần tử trong đối tượng $crawler và thực thi một hàm closure trên từng phần tử. Trả về một mảng chứa kết quả từng lần thực thi.

|  |
| --- |
| $crawler->filter('.product')->each(function ($node) {              // Xử lý từng phần tử ở đây  }); |

## text(): Trả về nội dung văn bản của phần tử đầu tiên trong đối tượng $crawler.

|  |
| --- |
| $crawler->filter('.product-name')->text(); // Trả về nội dung văn bản của phần tử có class "product-name" |

## attr($attribute): Trả về giá trị của một thuộc tính của phần tử đầu tiên trong đối tượng $crawler.

|  |
| --- |
| $crawler->filter('.product-image')->attr('src'); // Trả về giá trị của thuộc tính 'src' của phần tử có class "product-image" |

## count(): Đếm số lượng phần tử trong đối tượng $crawler.

|  |
| --- |
| $crawler->filter('.product')->count(); // Đếm số lượng phần tử có class "product" |

## first(): Trả về đối tượng Crawler mới chỉ chứa phần tử đầu tiên trong đối tượng $crawler.

|  |
| --- |
| $firstProduct = $crawler->filter('.product')->first(); // Lấy phần tử đầu tiên có class "product" |

## eq($index): Trả về đối tượng Crawler mới chỉ chứa phần tử ở vị trí chỉ định trong đối tượng $crawler.

|  |
| --- |
| $thirdProduct = $crawler->filter('.product')->eq(2); // Lấy phần tử thứ 3 có class "product" (vị trí 2) |

Tham khảo thêm tại [**GitHub - FriendsOfPHP/Goutte: Goutte, a simple PHP Web Scraper**](https://github.com/FriendsOfPHP/Goutte) , [The DomCrawler Component (Symfony Docs)](https://symfony.com/doc/current/components/dom_crawler.html)