Публикация на тему

Парсинг сайтов на Laravel

Реализация задач парсинга с помощью классов, интерфейсов, имплементов и типажей. Используется модуль Crowler фрэймворка Laravel

Автор
Михалькевич Александр Викторович

Публикация

Наименование Парсинг сайтов на Laravel

Автор А.В.Михалькевич

Специальность Реализация задач парсинга с помощью классов, интерфейсов, имплементов и типажей. Используется модуль Crowler фрэймворка Laravel,

Анотация

Anotation in English

Ключевые слова

Количество символов 17467

Содержание

Введение

1 Установка модуля dom-crawler

Сперва обновим composer

```
composer self-update
```

Установка модуля dom-crawler

```
composer require symfony/dom-crawler
```

После установки модуля, появится запись в файле composer.json

```
  "require": {
      "php": "^7.1.3",
      "fideloper/proxy": "^4.0",
      "laravel/framework": "5.6.*",
      "laravel/tinker": "^1.0",
      "laraveldaily/quickadmin": "^4.0",
      "symfony/dom-crawler": "^4.1"
  },
```

2 Интерфейс

В папке /app создаем еще одну папку /Parser, куда и помещаем интерфейс парсера

Содержимое файла ParseContract.php

```
namespace App\Parser;

Interface ParseContract
{
    public function getParse();
}
```

3 Типаж для парсинга

Создадим типаж с методоами text() html() и attr()

```
namespace App\Parser;
trait ParseTrait{
```

```
    public function text($obj, $val = null)
    {
        $risk = $obj->filter($val)->count();
        if ($risk == 0) {
            $rams = '';
        }else{
            $rams = $obj->filter($val)->text();
        }
        return $rams;
    }
    public function html($obj, $val = null)
    {
        $risk = $obj->filter($val)->count();
        if ($risk == 0) {
            $rams = '';
        }else{
            $rams = $obj->filter($val)->html();
        }
        return $rams;
    }
    public function attr($obj, $val=null, $atr=nul){
        $risk = $obj->filter($val)->count();
        if($risk == 0){
            $rams = '';
        }else{
            $rams = $obj->filter($val)->attr($atr);
        }
        return $rams;
    }
}
```

4 Имплемент парсинга с методом без входящих параметров

Для парсинга каталога товаров с сайта 24shop.by создадим класс Catalog.php в текущей папке /app/Parser

Отсутствие параметров для функции getParse() подсказывает то, что мы парсим все категории сайта.

```
namespace App\Parser;

use Symfony\Component\DomCrawler\Crawler;
//use App\ProductUser;
use App\Googlenew;
use Auth;

class Catalog implements ParseContract
```

```
{
    use ParseTrait;
    public $crawler;

    public function __construct()
    {
        set_time_limit(0);
        header('Content-Type: text/html; charset=utf-8');
    }

    public function getParse()
    {
        $ff = 'https://24shop.by/catalog/';
        $file = file_get_contents($ff);
        $this->crawler = new Crawler($file);
        //$tt = $this->html($this->crawler, '.images_table');
        $this->crawler->filter('.section__header')->each(function (Crawler
$node, $i) {
            $name = $this->text($node, "h3");
            $body = $this->text($node, ".esc-lead-snippet-wrapper");
            $picture = $this->attr($node, ".esc-thumbnail-image", "src");
            sleep(1);
            dd($name);
        });
    }
}
```

5 Имплемент для парсинга Onliner.by

Особенностью парсинга Onliner.by является то, что для парсинга мы используем api, предоставляющие данные в формате json. Доступны следующие ссылки для парсинга: `https://catalog.api.onliner.by/search/название_каталога`

Например:

> https://catalog.api.onliner.by/search/mobile
> https://catalog.api.onliner.by/search/printer

Список всех категорий сайта мы можем взять по ссылке - https://catalog.onliner.by

Сам имплемент выглядит так

```
namespace App\Parser;

use Symfony\Component\DomCrawler\Crawler;
use App\Parser\ParseContract;
use App\Product;
use App\Category;
use App\CatalogOnliner;
use Auth;
```

```
class Onliner implements ParseContract
{
    use ParseTrait;
    public $str = '';
    public $url = '';
    public $obj;
    public $char;
    public $crawler;
    public $id;

    public function __construct()
    {
        $file = file_get_contents('http://catalog.onliner.by/');
        $this->crawler = new Crawler($file);
    }

    public function getParse($data)
    {
        $divo = explode('/', $data->url);
        $end = end($divo);
        $pos = str_ireplace('http://catalog.onliner.by/', '', $data->url);
        $ask = strpos($pos, '?');
        if ($ask === false) {
            $vopros = '?';
        } else {
            $vopros = '&';
        }
        $http = 'https://catalog.api.onliner.by/search/' . $end . $vopros .
'page=1';
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $http);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $result = curl_exec($ch);
        curl_close($ch);
        $ke = 0;
        $data_arr = (array)json_decode($result);
        $cat = CatalogOnliner::where('name', $data->name)->first();
        $prod = Product::where('cat_id', $cat->id)->first();
        if (!is_null($prod)) {
            $prod->delete();
        }
        foreach ($data_arr as $key => $value) {
            if (is_array($value)) {

                foreach ($value as $val_key => $data_value) {
                    $val_value = (array)$data_value;
                    $site_id = (isset($val_value['id'])) ? $val_value['id'] :
'';
                    $site_key = (isset($val_value['key'])) ?
$val_value['key'] : '';
                    $name = (isset($val_value['full_name'])) ?
```

```
$val_value['full_name'] : '';
                    $pic_value = (array)$val_value['images'];
                    $header_picture = (isset($pic_value['header'])) ?
$pic_value['header'] : '';
                    $icon_picture = (isset($pic_value['icon'])) ?
$pic_value['icon'] : '';
                    $description = (isset($val_value['description'])) ?
$val_value['description'] : '';
                    $site_product_url = (isset($val_value['html_url'])) ?
$val_value['html_url'] : '';
                    $rev_value = (array)$val_value['reviews'];
                    $rating = (isset($rev_value['rating'])) ?
(integer)$rev_value['rating'] : '';
                    $site_reviews_url = (isset($rev_value['html_url'])) ?
$rev_value['html_url'] : '';
                    $price_value = (array)$val_value['prices'];
                    $price_min = (isset($price_value['min'])) ?
$price_value['min'] : '';
                    $price_max = (isset($price_value['max'])) ?
$price_value['max'] : '';
                    $site_prices_url = (isset($price_value['html_url'])) ?
$price_value['html_url'] : '';
                    $currency = (isset($price_value['currency_sign'])) ?
$price_value['currency_sign'] : '';
                    $forum_value = (array)$val_value['forum'];
                    $site_forum_url = (isset($forum_value['topic_url'])) ?
$forum_value['topic_url'] : '';
                    $site_api_url = (isset($val_value['url'])) ?
$val_value['url'] : '';
                    //получаем характеристики товара
                    $this->setCharacter($site_product_url, ".product-
specs__table");
                    $character = $this->getCharacter();
                    //добавляем товар в базу данных
                    $prod = new Product();
                    $prod->cat_id = (isset($cat->id)) ? $cat->id : 0;
                    $prod->site = 'onliner.by';
                    $prod->site_product_id = $site_id;
                    $prod->site_key = $site_key;
                    $prod->name = $name;
                    $prod->header_picture = $header_picture;
                    $prod->icon_picture = $icon_picture;
                    $prod->description = $description;
                    $prod->site_product_url = $site_product_url;
                    $prod->site_reviews_url = $site_reviews_url;
                    $prod->rating = $rating;
                    $prod->count = 1;
                    $prod->price_min = $price_min;
                    $prod->price_max = $price_max;
                    $prod->site_prices_url = $site_prices_url;
                    $prod->currency = $currency;
```

```
                    $prod->site_forum_url = $site_forum_url;
                    $prod->site_api_url = $site_api_url;
                    $prod->character = $character;
                    $prod->user_id = Auth::user()->id;
                    $prod->save();

                    sleep(1);
                    $ke++;
                }
            }
        }
        $body = $ke . " товаров";


        return $body;
    }


    public function catalog()
    {
        $arr = [];
        $this->crawler->filter(".catalog-navigation-
classifier__item")->each(function (Crawler $node, $i) {
            $id = ($node->attr('data-id') != null) ? $node->attr('data-id') :
0;
            $cat = CatalogOnliner::firstOrNew(['id' => $id]);
            $cat->name = $node->filter('.catalog-navigation-classifier__item-
title-wrapper')->text();
            $cat->type = 'link';
            $cat->parent_id = 0;
            $cat->save();
        });
        return $arr;
    }

    public function setCharacter($path, $selector)
    {
        $file = file_get_contents($path);
        $crawler = new Crawler($file);
        $table = $this->html($crawler, $selector);
        $this->char = $table;
        return true;
    }

    public function h2Catalog()
    {
        $parent = CatalogOnliner::all();
        foreach ($parent as $one) {
            $this->id = $one->id;
            $this->crawler->filter(".catalog-navigation-list__category[data-
id=" . $one->id . "]")->each(function (Crawler $node, $i) {
```

```
                $node->filter('.catalog-navigation-list__group-
title')->each(function (Crawler $node, $i) {
                    //echo $node->text();
                    $cat = new CatalogOnliner;
                    $cat->name = $node->text();
                    $cat->parent_id = $this->id;
                    $cat->type = 'h2';
                    $cat->save();
                });

            });
        }
    }

    public function listCatalog()
    {
        $parent = CatalogOnliner::all();
        foreach ($parent as $one) {
            $this->id = $one->id;
            $this->crawler->filter(".catalog-navigation-list__category[data-
id=" . $one->id . "]")->each(function (Crawler $node, $i) {
                $node->filter('.catalog-navigation-
list__group')->each(function (Crawler $node, $i) {
                    $cat = new CatalogOnliner;
                    $cat->name = $node->filter('.catalog-navigation-
list__group-title')->text();
                    $cat->category_type = 'h2';
                    $cat->parent_id = $this->id;
                    $cat->type = 'h2';
                    //$cat->save();

                    $node->filter('.catalog-navigation-list__link-inner
a')->each(function (Crawler $node, $i) {
                        $cat = new CatalogOnliner;
                        $cat->name = $node->text();
                        $cat->url = $node->attr('href');
                        $cat->parent_id = $this->id;
                        $cat->type = '';
                        //$cat->save();
                    });
                });

            });
        }


    }

    public function getCharacter()
    {
        return $this->char;
```

```
    }
}
```

6 Имплемент для парсинга wikipedia.org

Исходный код имплемента для парсинга wikipedia.org

```
namespace App\Parser;

use Symfony\Component\DomCrawler\Crawler;
//use App\ProductUser;
//use App\Googlenew;
use Auth;

class WikipediaInfo implements ParseContract
{
    use ParseTrait;
    public $crawler;

    public function __construct()
    {
        set_time_limit(0);
        header('Content-Type: text/html; charset=utf-8');
    }

    public function getParse($country="Belarus")
    {
        $ff = 'https://ru.wikipedia.org/wiki/'.$country;
        $file = file_get_contents($ff);
        $this->crawler = new Crawler($file);
        //$tt = $this->html($this->crawler, '.images_table');
                $body=$this->crawler->filter('body')->html();
                echo $body;
    }
}
```

7 Имплемент для парсинга news.google.com

```
namespace App\Parser;

use Symfony\Component\DomCrawler\Crawler;
use App\News;
use Auth;

class GoogleNews implements ParseContract
{
    use ParseTrait;
    public $crawler;
```

```
    public function __construct()
    {
        set_time_limit(0);
        header('Content-Type: text/html; charset=utf-8');
    }

    public function getParse($catalog="Belarus", $cat_id=null)
    {
        $ff =
'https://news.google.com/search?q='.$catalog.'&hl=en-US&gl=US&ceid=US%3Aen';
        $file = file_get_contents($ff);
        $this->crawler = new Crawler($file);
        //$tt = $this->html($this->crawler, '.images_table');
        $this->crawler->filter('.section')->each(function (Crawler $node, $i)
{
            $name = $this->text($node, "h3");
            $body = $this->text($node, ".esc-lead-snippet-wrapper");
            $picture = $this->attr($node, ".esc-thumbnail-image", "src");
            $obj = new News;
            $obj->name = $name;
            $obj->body = $body;
            $obj->picture = $picture;
            $obj->catalog = $cat_id;
            $obj->user_id = (Auth::guest())?0:Auth::user()->id;
            $obj->save();
            sleep(1);
        });
    }
}
```

8 Парсинг

На каждый выбранный ресуср парсинга (сайт) создаем свои кнопки, Это может быть как и обычная html-ссылка, так и любой элемент , к которому добавлен прослушиватель.

Для каждого сайта свой прослушиватель с вызовом ajax.

```
        $(document).ready(function () {
            $('#catalog_24_shop').click(function () {
                $.ajax({
                    type: "Post",
                    url: '/ajax/parse/catalog',
                    data: 'parse=ok',
                    success: function (data) {
                        $('#empty_catalog').html(data);
                    }
                });
            });
        });
```

8 .1 Контроллер парсинга и маршруты на него

Для парсинга каждого сайта нужен свой маршрут, но мы можем обойтись одним контроллером. Поэтому, сперва создадим контроллер:

```
php artisan make:controller ParseController
```

В файле routes/web.php прописываем маршруты на контроллер ParseController

```
Route::post('ajax/parse/onliner', 'ParseConroller@postOnliner');


Route::post('ajax/parse/wikipedia', 'ParseConroller@postWikipedia');
```

8 .2 Экшн парсинга Wikipedia

```
        public function getInfo(){
                $country=$_GET["country"];
                $obj=new Wikipedia;
                $pars=$obj->getParse($country);
     return view('ajax.info',compact("country"));
        }
```

8 .3 Экшн парсинга news.google.com

Парсим новости выбранной страны только в случае первого вызова экшна. Т.к. мы не хотим парсить новости которые у нас уже есть, перед парсингом делаем проверку по дате последнего обращения к странице новостей.

```
    public function getNews($country = null){
                //$country=$_GET["country"];
                $day = date('Y-m-d');

                $ob = News::where('country_name', $country)->where('putday',
$day)->first();
                if(!$ob){
            $obj=new GoogleNews;
            $pars=$obj->getParse($country);
            $obn  = new News;
            $obn->country_name = $country;
            $obn->body = $pars;
            $obn->lang = 'en';
            $obn->putday = $day;
            $obn->url = $_SERVER['SERVER_NAME'].$_SERVER['REQUEST_URI'];
            $obn->save();
            $parse = $pars;
        }else{
$parse = $ob->body;
```

```
        }
     return view('ajax.iframe',compact("country", "parse"));
    }
```

Этот код использует модель GoogleNews, которую нужно подключить с помощью конструкции use
Информация по созданию и подключению моделей доступна по ссылке
http://erud.by/object_orient_program/627

Заключение

Список использованных источников

Приложения