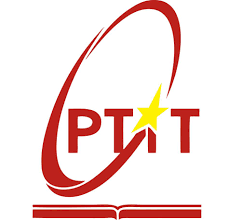
****

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO THỦ THUẬT PHP**

****

**THỦ THUẬT: Cách tạo router trong PHP**

**Giáo viên: Nguyễn Quốc Hưng**

**Môn: Lập trình web - Nhóm 03**

**Sinh viên thực hiên: Dương Bằng Huân - B18DCCN251**

**Hà Nội, ngày 13 tháng 11 năm 2021**

**Mục lục**

**[Lơi ích sử dụng router: 3](#_Toc11355)**

**[Mục tiêu: 3](#_Toc17243)**

**[Cài đặt: 3](#_Toc3718)**

[1. Tạo ra project có cấu trúc thư mục: 3](#_Toc23410)

[2. Thực hiện cấu hình file .htaccess 3](#_Toc3218)

[3. Thực hiện viết core xử lý router (core/http/Route.php) 5](#_Toc2041)

[4. Chạy thử 10](#_Toc9759)

# Lơi ích sử dụng router:

- Url của trang web sẽ đẹp hơn và người dùng có thể dễ dàng ghi nhớ.

- Url sẽ tối ưu hơn và chuẩn SEO hơn (Search Engine Optimization) và sẽ đạt thứ hạng cao hơn trên các công cụ tìm kiếm.

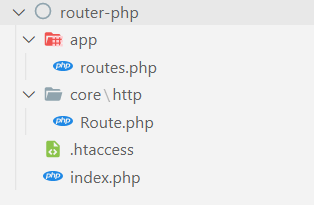
# Mục tiêu:

- Tạo ra library router giống với các Framework (Laravel, Codeigniter,…) bằng PHP thuần

- Hiểu được cơ chế tạo route trong các framework PHP

# Cài đặt:

## Tạo ra project có cấu trúc thư mục:



Trong đó:

- **index.php**: File endpoint của ứng dụng

- **.htaccess**: File cấu hình apache rewrite url

- **core/http/Route.php**: Core thực hiện điều hướng url đến đúng route của ứng dụng

- **app/routes.php**: file thiết lập các route của ứng dụng

## Thực hiện cấu hình file .htaccess

RewriteEngine On

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteRule ^(.+)$ index.php

Chức năng của file trên là: gọi url được gọi trên server đều sẽ truy cập vào file **index.php**

Tiếp theo chúng ta cần thêm đoạn code bắt được query string url trên trong file **index.php**

<?php

// Định nghĩa hằng Path của file index.php

define('PATH\_ROOT', \_\_DIR\_\_);

// Autoload class trong PHP

spl\_autoload\_register(function (string $class\_name) {

    include\_once PATH\_ROOT . '/' . $class\_name . '.php';

});

// lấy giá trị của script\_name

$script\_name = !empty($\_SERVER['SCRIPT\_NAME']) ? $\_SERVER['SCRIPT\_NAME'] : "/index.php";

$script\_name = str\_replace("/index.php", "", $script\_name);

// lấy giá trị của request\_uri

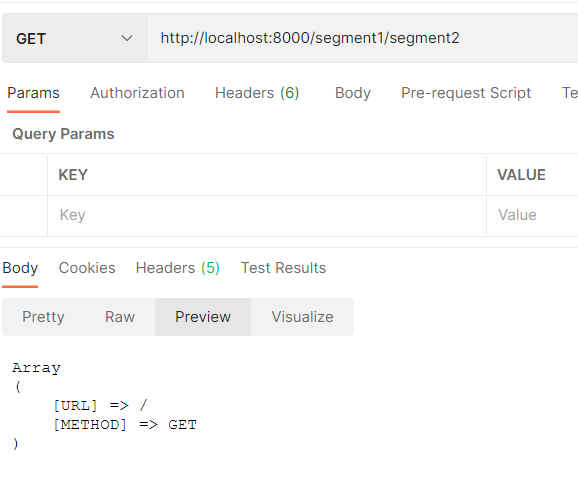
$request\_uri = !empty($\_SERVER['REQUEST\_URI']) ? $\_SERVER['REQUEST\_URI'] : '/';

// Lấy url hiện tại của trang web. Mặc định la /

$request\_url = str\_replace($script\_name, "", $request\_uri);

// Lấy phương thức hiện tại của url đang được gọi. (GET | POST). Mặc định là GET.

$method\_url = !empty($\_SERVER['REQUEST\_METHOD']) ? $\_SERVER['REQUEST\_METHOD'] : 'GET';



## Thực hiện viết core xử lý router (core/http/Route.php)

Ta xây dựng bộ khung cho phần core trước

<?php

namespace Core\Http;

/\*\*

 \*

 \* Class Route

 \*

 \*/

class Route

{

    /\*\*

     \*

     \* - Mảng lưu trữ route của ứng dụng

     \* - Mỗi route sẽ gôm url, method, action và params

     \*

     \*/

    private $\_\_routes;

    // Hàm khởi tạo

    public function \_\_construct()

    {

        $this->\_\_routes = [];

    }

    /\*\*

     \*

     \* Phương thức get

     \*

     \* @param string $url URL cần so khớp

     \* @param string|callable $action Hành động khi URL được gọi. Có thể là một callback hoặc một method trong controller

     \*

     \* @return void

     \*

     \*/

    public function get(string $url, $action)

    {

            // Xử lý phương thức GET

            $this->\_\_request($url, 'GET', $action);

    }

    /\*\*

     \*

     \* Phương thức POST

     \*

     \* @param string $url URL cần so khớp

     \* @param string|callable $action Hành động khi URL được gọi. Có thể là một callback hoặc một method trong controller

     \*

     \* @return void

     \*

     \*/

    public function post(string $url, $action)

    {

         // Xử lý phương thức POST

         $this->\_\_request($url, 'POST', $action);

    }

    /\*\*

     \*

     \* Xử lý phương thức

     \*

     \* @param string $url URL cần so khớp

     \* @param string $method method của route. GET hoặc POST

     \* @param string|callable $action Hành động khi URL được gọi. Có thể là một callback hoặc một method trong controller

     \*

     \* @return void

     \*

     \*/

    private function \_\_request(string $url, string $method, $action)

    {

        // kiem tra xem URL co chua param khong. VD: post/{id}

        if (preg\_match\_all('/({([a-zA-Z]+)})/', $url, $params)) {

            $url = preg\_replace('/({([a-zA-Z]+)})/', '(.+)', $url);

        }

        // Thay the tat ca cac ki tu / bang ky tu \/ (regex) trong URL.

        $url = str\_replace('/', '\/', $url);

        $route = [

            'url' => $url,

            'method' => $method,

            'action' => $action,

            'params' => $params[2]

        ];

        array\_push($this->\_\_routes, $route);

    }

    /\*\*

     \*

     \* Hàm xử lý khi một URL được gọi

     \*

     \* @param string $url URL được gọi đến server

     \* @param string $method Phương thức url được gọi. GET | POST

     \*

     \* @return void

     \*

     \*/

    public function map(string $url, string $method)

    {

            // code

    }

}

Các hàm trên đã em đã comment chức năng của hàm đó

Bây giờ em sẽ thực hiện code cho từng hàm đó. Bắt đầu với hàm request

private function \_\_request(string $url, string $method, $action)

    {

        // kiểm tra xem URL có chứa param không. VD: post/{id}

        if (preg\_match\_all('/({([a-zA-Z]+)})/', $url, $params)) {

            // thay thế param bằng (.+). VD: post/{id} -> post/(.+)

            $url = preg\_replace('/({([a-zA-Z]+)})/', '(.+)', $url);

        }

        // Thay thế tất cả các kí tự / bằng ký tự \/ (regex) trong URL.

        $url = str\_replace('/', '\/', $url);

        // Tạo một route mới

        $route = [

            'url' => $url,

            'method' => $method,

            'action' => $action,

            'params' => $params[2]

        ];

        // Thêm route vào router.

        array\_push($this->\_\_routes, $route);

    }

Quay lại file **index.php** thêm dòng code

// map URL

$router->map($request\_url, $method\_url);

Dòng code sẽ thực hiện map url với route trong ứng dụng để xác định client đang yêu cầu đến route nào

Quay lại file **core/http/Route.php**, em viết hàm map

/\*\*

     \*

     \* Hàm xử lý khi một URL được gọi

     \*

     \* @param string $url URL được gọi đến server

     \* @param string $method Phương thức url được gọi. GET | POST

     \*

     \* @return void

     \*

     \*/

    public function map(string $url, string $method)

    {

        // Lặp qua các route trong ứng dụng, kiểm tra có chứa url được gọi không

        foreach ($this->\_\_routes as $route) {

            // nếu route có $method

            if ($route['method'] == $method) {

                // kiểm tra route hiện tại có phải là url đang được gọi.

                $reg = '/^' . $route['url'] . '$/';

// loại bỏ chuỗi query string trong url (nếu có)

                $querystring = strpos($url, "?");

                if($querystring != false){

               $url = substr($url, 0, $querystring);

          }

                if (preg\_match($reg, $url, $params)) {

                    array\_shift($params);

                    $this->\_\_call\_action\_route($route['action'], $params);

                    return;

                }

            }

        }

        // nếu không khớp với bất kì route nào cả.

        echo '404 - Not Found';

        return;

    }

    /\*\*

     \*

     \* Hàm gọi action route

     \*

     \* @param string|callable $action action của route

     \* @param array $params Các tham số trên url

     \*

     \* @return void

     \*

     \*/

    private function \_\_call\_action\_route($action, $params)

    {

        // Nếu $action là một callback (một hàm).

        if (is\_callable($action)) {

            call\_user\_func\_array($action, $params);

            return;

        }

        // Nếu $action là một phương thức của controller. VD: 'HomeController@index'.

        if (is\_string($action)) {

            $action = explode('@', $action);

            $controller\_name = 'App\\Controllers\\' . $action[0];

            $controller = new $controller\_name();

            call\_user\_func\_array([$controller, $action[1]], $params);

            return;

        }

    }

Vậy là đã hoàn thành phần core của router trong php, giờ có thể sang file **app/routes.php** cấu hình thử một route

## Chạy thử

<?php

// action là callback

$router->get('/', function(){

    echo 'home';

});

Ngoài ra, ta có thể tạo một class controller trong thư mục app/controllers. Ví dụ:

<?php

namespace App\Controllers;

class HomeController {

    public function index(){

        echo "home";

    }

}

Và cấu hình lại file **app/routes.php** theo controller

// action là method trong class

$router->get('/', "HomeController@index");