

Laravel

Ths. Ngô Lê Quân
ngolequanit@gmail.com
www.Vku.udn.vn

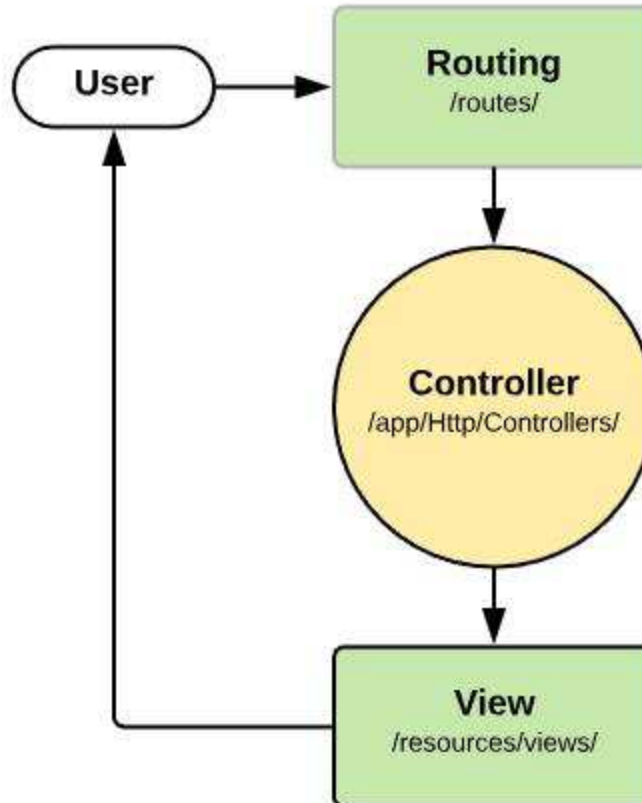
Bài 5. Controller

5.1. Controller là gì?

- Để xử lý logic (tính toán) thì chúng ta có thể viết trong nhiều chỗ khác nhau, nhưng tốt nhất là gom chung viết bên trong Controller, đặc biệt là các tính toán dài, phức tạp.
- Controller sinh ra là để trở thành trung tâm xử lý logic, nên cần phải tận dụng, thay vì viết tính toán ở nơi khác.
- Controller được đặt bên trong thư mục */app/Http/Controllers/*.
- Để hiểu một cách cơ bản nhất cách hoạt động của một Controller ta xem cấu trúc đơn giản sau:

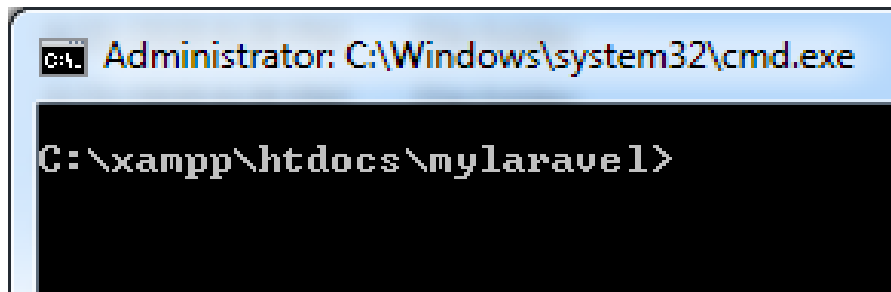
Controller là gì?

- Để hiểu một cách cơ bản nhất cách hoạt động của một Controller ta xem cấu trúc đơn giản sau:



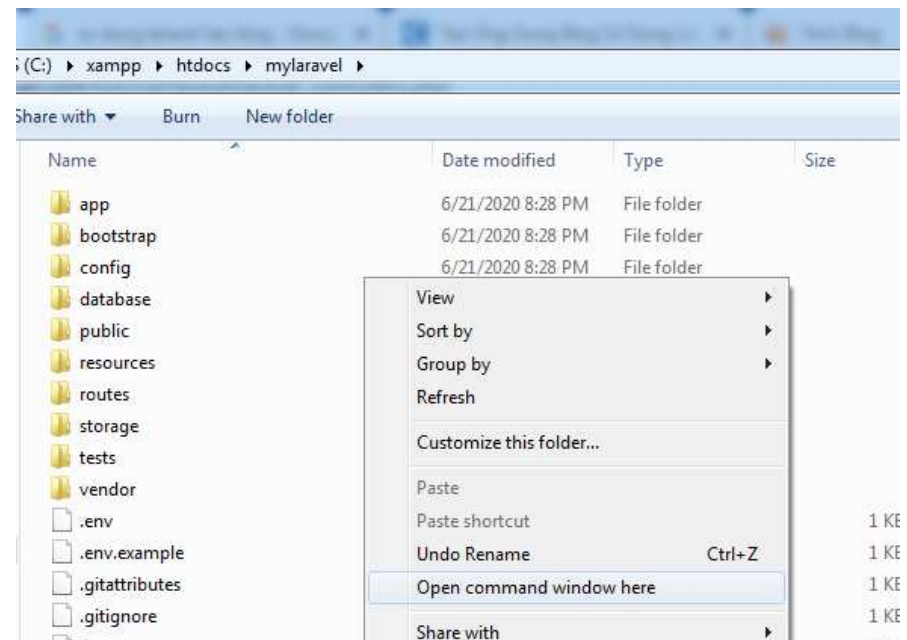
Tạo một Controller

- Trước tiên, mở cửa sổ lệnh cmd, di chuyển tới thư mục mylaravel(website đã tạo) bằng lệnh:
- Nhấn shift và nhấp phải chuột để chọn mục **Open command window here**



```
Administrator: C:\Windows\system32\cmd.exe

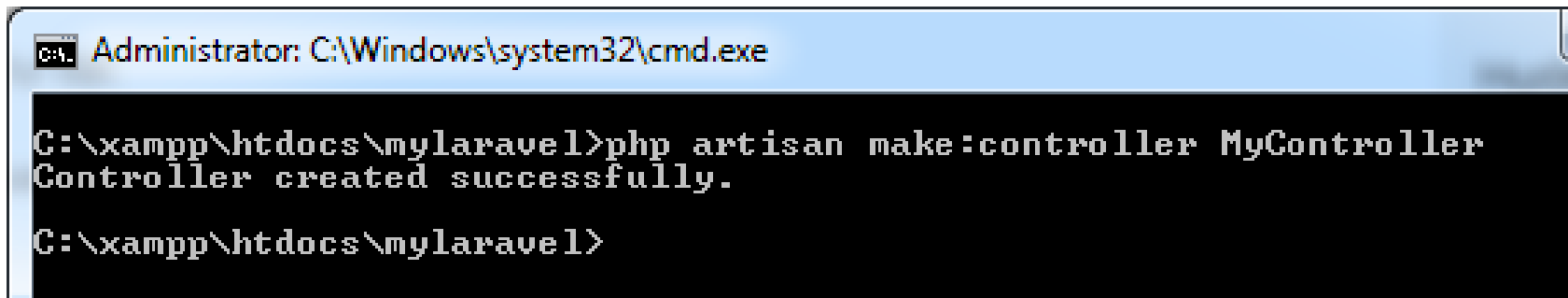
C:\xampp\htdocs\mylaravel>
```



Tạo một Controller

- Tạo Controller bằng lệnh Artisan

```
php artisan make:controller MyController
```



```
Administrator: C:\Windows\system32\cmd.exe  
  
C:\xampp\htdocs\mylaravel>php artisan make:controller MyController  
Controller created successfully.  
  
C:\xampp\htdocs\mylaravel>
```

Tạo một Controller

- Nếu xuất hiện thông báo "Controller created successfully." thì quá trình tạo file thành công.
- Lúc này xem trong thư mục `/app/Http/Controllers/` sẽ có thêm file `MyController.php` với nội dung bên trong mặc định như sau:

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class MyController extends Controller
{
    //
}
```

Chạy một Controller

- Thực hiện hiển thị một trang Views từ việc điều khiển phần Routing gửi yêu cầu tới Controller xử lý, xong kết quả sẽ trả lại về Views.
- Viết nội dung file Controller *MyController.php* vừa tạo với nội dung sau:

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class MyController extends Controller
{
    public function index()
    {
        return view('test'); // trả về trang test.blade.php
    }
}
```

Chạy một Controller

- Tạo thêm một file View *test.blade.php* đặt bên trong thư mục */resources/views/*, với nội dung:

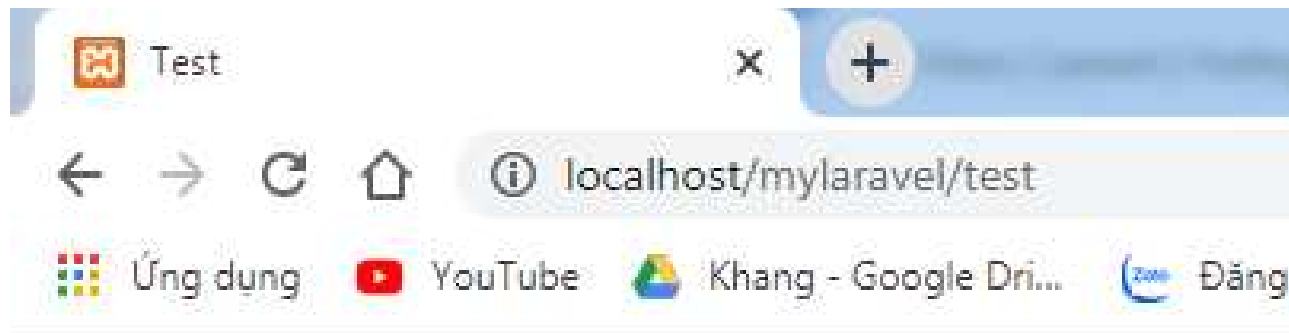
```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Test</title>
</head>
<body>
Đây là trang Test
</body>
</html>
```


Chạy một Controller

- Viết một routing `/routes/web.php` đơn giản hiển thị trang view `/resources/views/test.lade.php` thông qua một controller:

```
Route::get('/test', 'MyController@index');
```

- Gõ trên trình duyệt để kiểm tra kết quả:



Đây là trang Test

Xử lý biến với Controller

- Mở file MyController và thêm vào lệnh:

return view('test')->with('title', \$title);

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class MyController extends Controller
{
    public function index()
    {
        $title = "Đây là tiêu đề";
        return view('test')->with('title', $title);
    }
}
```

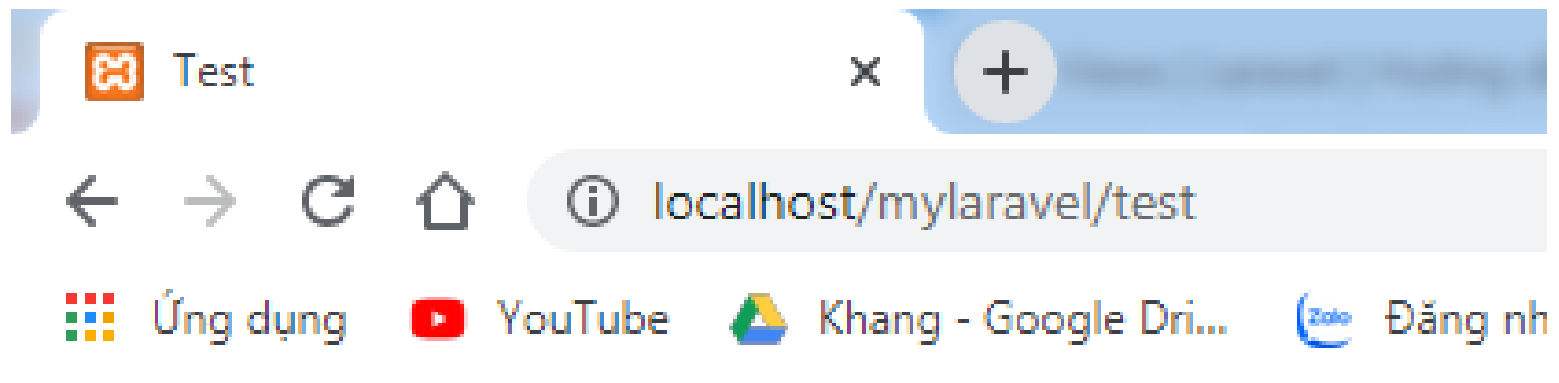
Xử lý biến với Controller

➤ Để nhận giá trị biến ta viết lại *test.blade.php* như sau:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Test</title>
</head>
<body>
<h2>{{ $title }}</h2>
Đây là trang Test
</body>
</html>
```

Xử lý biến với Controller

- Hiển thị kết quả trên trình duyệt:



Đây là tiêu đề

Đây là trang Test

Xử lý nhiều biến với Controller

- Mở file MyController và sửa lại hàm index() như sau:

```
public function index()
{
    $title = "Đây là tiêu đề";
    $description = "Đây là dòng mô tả";
    $copyright = "Học Web SDC";
    return view('test')->with(['title'=>$title,
    'description'=>$description,
    'copyright'=>$copyright]);
}
```

Xử lý nhiều biến với Controller

➤ Để nhận giá trị các biến ta viết lại *test.blade.php* như sau:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Test</title>
</head>
<body>
<h2>{{ $title }}</h2>
<div>{{ $description }}</div>
Đây là trang Test
<p>{{ $copyright }}</p>
</body>
</html>
```