

Bài giảng

PHÁT TRIỂN ỨNG DỤNG WEB

Lê Đình Thanh

Khoa Công nghệ Thông tin

Trường Đại học Công nghệ, ĐHQGHN

E-mail: thanhld@vnu.edu.vn Mobile: 0987.257.504

Laravel

Nội dung

- Tổng quan
- Định tuyến
- Tạo controller
- Tạo model
- Tạo view

Framework là gì?

- Khung phát triển ứng dụng (software framework - SF) là phần mềm cung cấp các chức năng chung có thể thay đổi bởi mã của người dùng.
- SF bao gồm:
 - Một tập các thành phần phần mềm có thể dễ dàng tích hợp với nhau
 - Một phương pháp lắp ráp các thành phần phần mềm và viết mã bổ sung

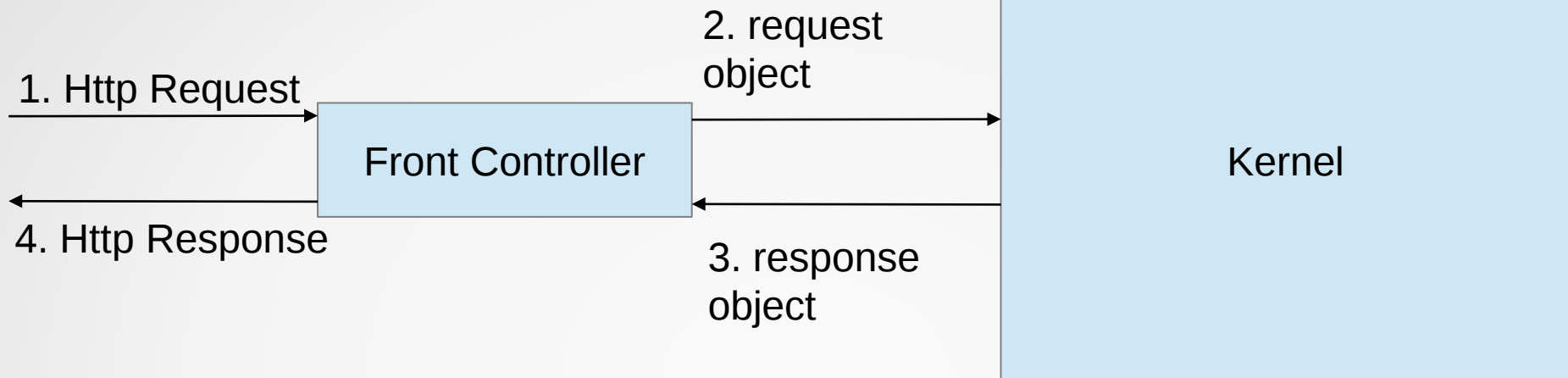
Lợi ích của việc sử dụng SF

- Viết ít mã, với ít lỗi, và đảm bảo khả năng bảo trì, nâng cấp
 - Phát triển ứng dụng đúng (chất lượng tốt) và hiệu quả (năng suất cao)

Laravel

- Là khung phát triển ứng dụng web với PHP
 - nguồn mở
 - được phát triển một cộng đồng lớn
 - được sử dụng rộng rãi
- Websites:
 - <https://laravel.com/>

Luồng xử lý



- Bộ điều khiển mặt trước (front controller) khởi động nhân ứng dụng (kernel), tiếp nhận Http Request, tạo đối tượng request và đưa Http Request vào đối tượng request được tạo, sau đó chuyển đối tượng request cho nhân
- Nhân xử lý request, trả kết quả là đối tượng response cho bộ điều khiển mặt trước
- Tại bộ điều khiển mặt trước, đối tượng response được chuyển thành Http Response rồi gửi cho client

Luồng xử lý

//Khởi động nhân ứng dụng

`$kernel = $app-`

`>make(Illuminate\Contracts\Http\Kernel::class);`

//Tạo đối tượng request nắm giữ Http Request

`$request = Illuminate\Http\Request::capture();`

//Chuyển đối tượng request cho nhân xử lý, nhận kết quả xử lý là đối tượng response

`$response = $kernel->handle($request);`

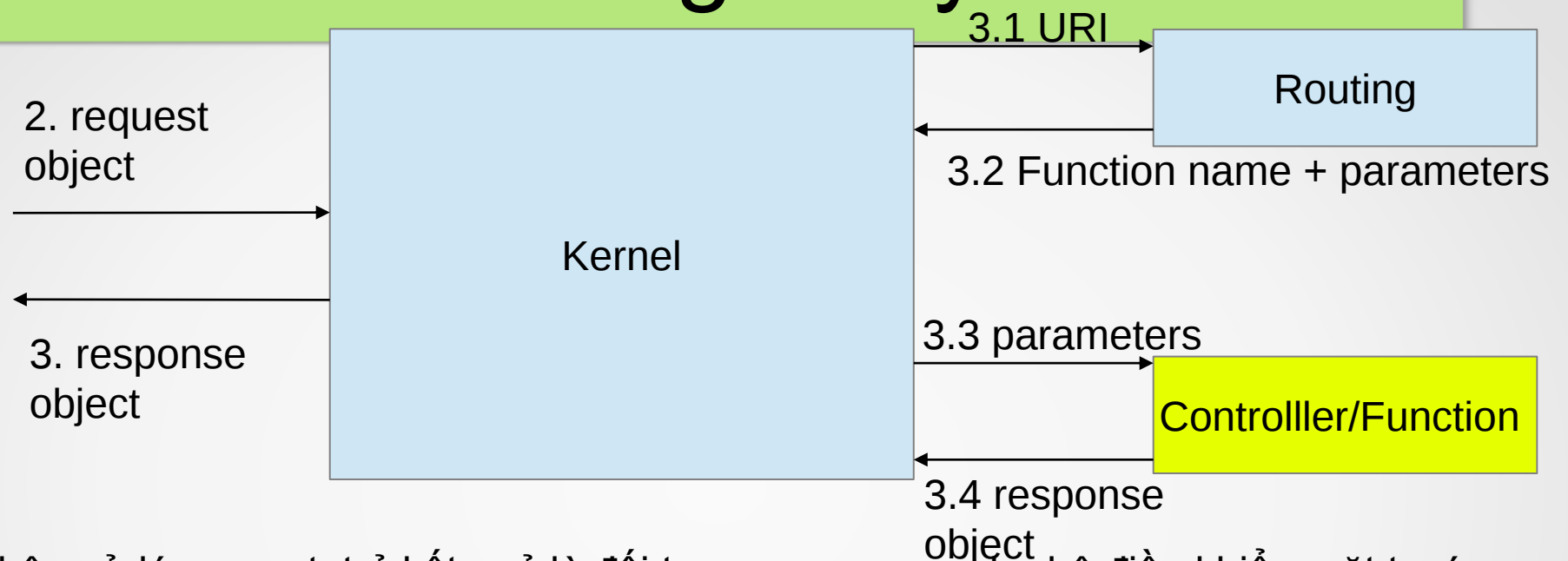
//Tạo và gửi Http Response cho client

`$response->send();`

//Tắt nhân

`$kernel->terminate($request, $response);`

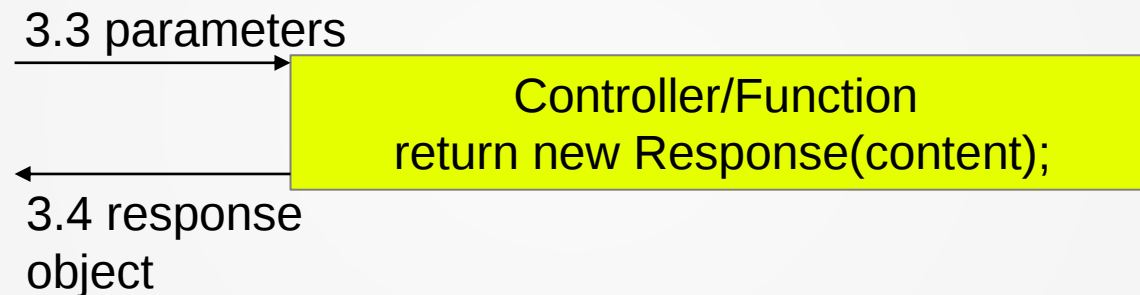
Luồng xử lý



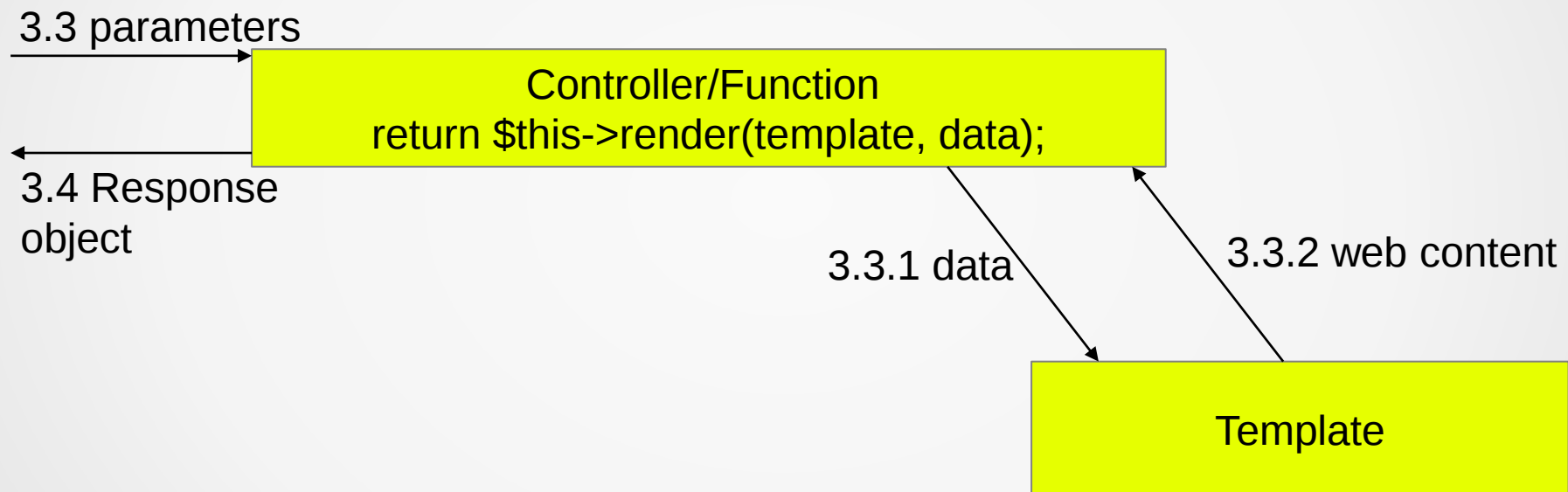
- Nhân xử lý request, trả kết quả là đối tượng response cho bộ điều khiển mặt trước
 - **Nhân chuyển URI của request cho hệ thống định tuyến (routing)**
 - **Hệ thống định tuyến xác định hàm PHP (còn được gọi là bộ điều khiển (controller) hay hành động (action)) nào sẽ được gọi**
 - **Hàm PHP được gọi thực hiện thông dịch, xử lý request và tạo đối tượng response**
 - **Đối tượng response được trả về cho nhân và nhân chuyển cho bộ điều khiển mặt trước**

Tạo “trang web” = viết hàm xử lý request và tạo response + ánh xạ URL tới hàm

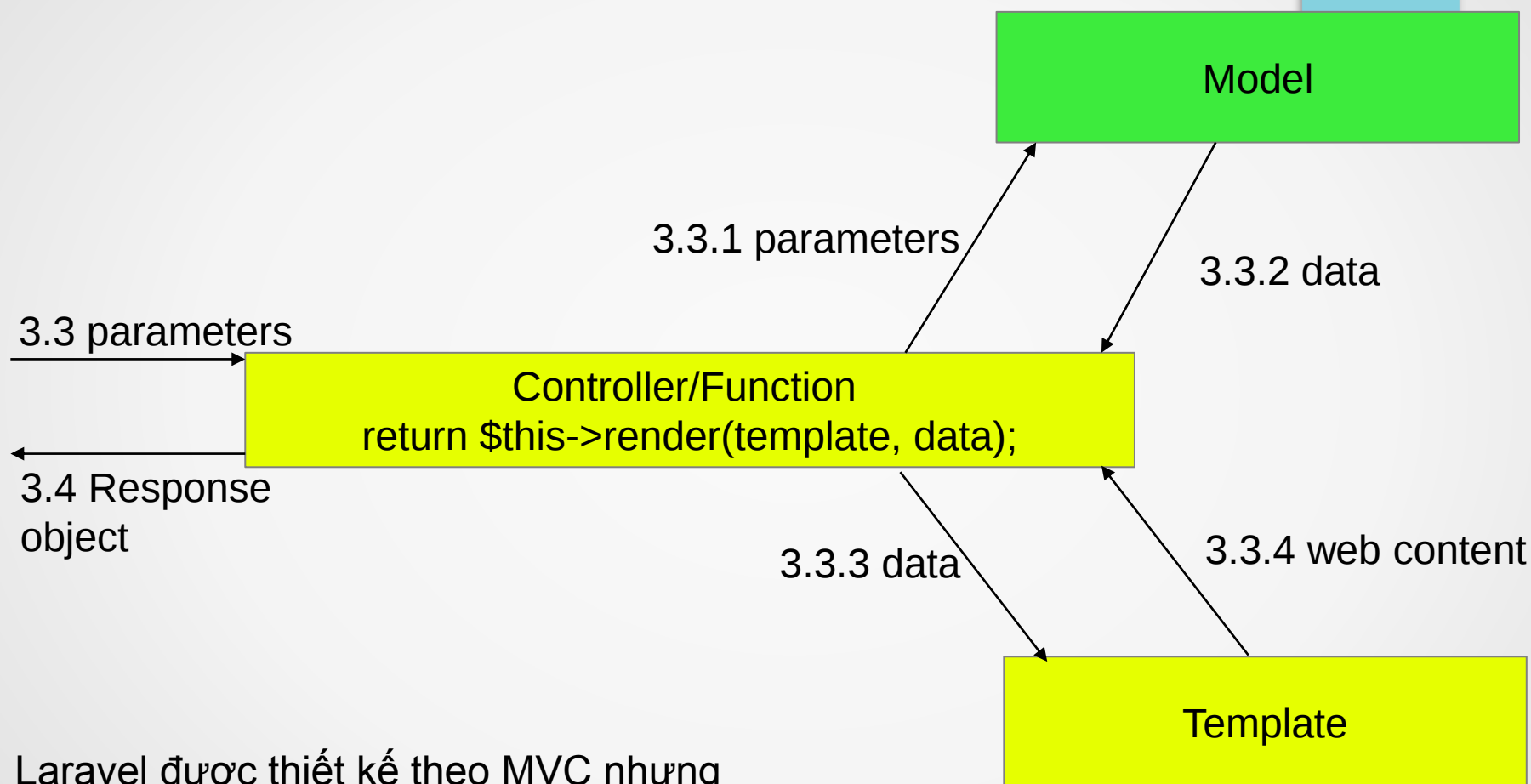
Controller trực tiếp tạo Response



Controller sử dụng template/view để tạo Response



Sử dụng model



Laravel được thiết kế theo MVC nhưng không bắt buộc và không hỗ trợ model. Người phát triển tự tạo model.

Cấu trúc thư mục

- *app/Http/Controllers*
chứa các tệp cài đặt các lớp điều khiển
- *resources/views*
chứa các tệp cài đặt các lớp giao diện
- *app/Models*
chứa các các tệp cài đặt các lớp mô hình
- *routes*
chứa các tệp lưu trữ thông tin cho định tuyến URL
- *config*
lưu các tệp cấu hình của ứng dụng,
- *database*
lưu các tệp liên quan đến thao tác CSDL
- *public*
lưu các tài nguyên khác như hình ảnh, JavaScript, CSS, ...
- *storage*
lưu các tệp session, cache và các tệp khác được sinh ra bởi khung làm việc
- *tests*
chứa các kịch bản kiểm thử do người lập trình tạo lập
- *vendor*
chứa các môđun do bên thứ ba cung cấp.

Tạo ứng dụng mới

- Sử dụng Composer (<https://getcomposer.org>)

*composer create-project laravel/laravel <AppName>
--prefer-dist*

Thiết lập thông tin định tuyến URL

- *Route::method(\$uri, \$callback);*

method = HTTP verb: get, post, put, delete, ...

- Ví dụ

```
Route::get('/greeting', function() { return view('welcome'); });
```

```
Route::get('readers/{readerId}', function ($rid) {return 'Độc giả: '.$rid; });
```

```
Route::get('posts/{postId}/comments/{commentId}', function($pid, $cid ) { return 'Bạn đang đọc bình luận '.$cid.' ở bài viết'.$pid; });
```

```
Route::match(['get', 'post'], '/help', function () { return view('help'); });
```

```
Route::any('/licence', function () { return "Apache Licence"; });
```

```
Route::redirect('/here', '/there', 301);
```

Xây dựng lớp điều khiển

- `php artisan make:controller <ControllerName>`

- Ví dụ

php artisan make:controller BookController

```
<?php
// Tập app/Http/Controllers/BookController.php
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;

class BookController extends Controller
{
    public function show($bid)
    {
        return "Xem thông tin cuốn sách $bid";
    }
}
```

Ánh xạ URL đến hành động của lớp điều khiển

```
<?php
```

```
// Tập routes/web.php
```

```
...
```

```
Route::get('/books/{bookId}', 'BookController@show');
```

Ánh xạ URL đến hành động của lớp điều khiển

Route::resource('/books, 'BookController');

Phương thức HTTP	URI	Hành động
<i>GET</i>	<i>/books</i>	<i>index</i>
<i>GET</i>	<i>/books/create</i>	<i>create</i>
<i>POST</i>	<i>/books</i>	<i>store</i>
<i>GET</i>	<i>/books/{bookId}</i>	<i>show</i>
<i>GET</i>	<i>/books/{bookId}/edit</i>	<i>edit</i>
<i>PUT/PATCH</i>	<i>/books/{bookId}</i>	<i>update</i>
<i>DELETE</i>	<i>/books/{bookId}</i>	<i>destroy</i>

Đọc giá trị của tham số trong yêu cầu HTTP

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;

class BookController extends Controller {
    //
    public function store(Request $request) {
        $title = $request->input('title');
        $author = $request->input('authors');
        //
    }
    public function update(Request $request, $bid) {
        $title = $request->input('title');
        $author = $request->input('authors');
        //
    }
}
```

Trả về text hoặc JSON

```
class BookController extends Controller {  
  // Trả về text  
  public function index() {  
    return 'Sách: Gió Thu, Sóng Sánh, Chiều Hồng';  
  }  
  // Trả về JSON  
  public function index2() {  
    return ['Gió Thu', 'Sóng Sánh', 'Chiều Hồng'];  
  }  
}
```

Trả về đối tượng response

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Response;

class BookController extends Controller {
//
    public function index() {
        $content = ['Gió Thu', 'Sóng Sánh', 'Chiều Hồng'];
        return response($content) ;
    }
}
```

Thêm tiêu đề cho HTTP response

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Response;

class BookController extends Controller {
//
public function index() {
    $content = ['Gió Thu', 'Sóng Sánh', 'Chiều Hồng'];
    return response($content)
        ->withHeaders([
            'Content-Type' => 'application/json',
            'Set-Cookie' => 'view=list;HttpOnly'
        ]);
    }
}
```

Tạo model

php artisan make:model Models\Book

```
<?php
// Tập app/Models/Book.php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;

class Book extends Model {
    protected $table='Sach';
}
```

Tạo view

```
<!-- resources/views/layouts/master.blade.php -->
<!DOCTYPE html><html><head>
  <title>ABC website</title>
  <meta charset="utf-8">
</head><body>
  <div class="container">
    <div>@yield('header')</div>
    <div class="row">
      <div class="col-md-3">
        @section('leftmenu')
          Left menu
        @show
      </div>
      <div class="col-md-9">@yield('content')</div>
    </div>
  </div>
</body></html>
```

Tạo view

```
<!-- resources/views/layouts/home.blade.php -->
```

```
@extends('layouts.master')
```

```
@section('header')
```

```
<p>Welcome.</p>
```

```
@endsection
```

```
@section('leftmenu')
```

```
@parent
```

```
<ul class="vertical-menu">
```

```
<li><a href="#1">Menu 1</a></li>
```

```
<li><a href="#2">Menu 2</a></li>
```

```
</ul>
```

```
@endsection
```

```
@section('content')
```

```
<p>ABC website.</p>
```

```
@endsection
```

Truyền dữ liệu cho view

```
<?php
// Tập app/Http/Controllers/BookController.php
namespace App\Http\Controllers;
use App\Book;
use App\Http\Controllers\Controller;
class BookController extends Controller {
    public function show($id) {
        return view('showbook', ['book' => Book::find
OrFail($id)]);
    }
}
```

Truyền dữ liệu cho view

```
<!-- resources/views/layouts/showbook.blade.php -->
```

```
@extends('layouts.master')
```

```
@section('header')
```

```
<p>Thông tin sách</p>
```

```
@endsection
```

```
@section('content')
```

```
<h2>{{ $book->tieude }} </h2>
```

```
Các tác giả:
```

```
@foreach ($book->authors as $author)
```

```
    {{ $author->hoten }} <br>
```

```
@endforeach
```

```
Năm xuất bản: {{ $book->namxb }}
```

```
@endsection
```


Tiếp theo
Xu hướng phát triển web