

Buổi 8

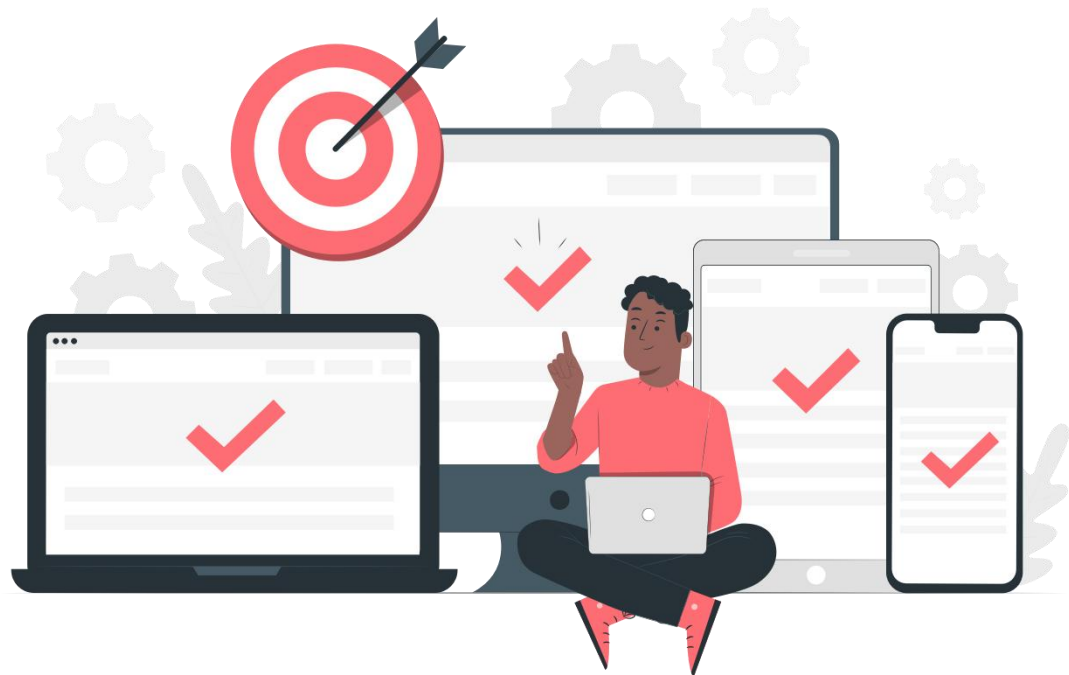
Mô hình MVC

Laravel Route - Laravel Blade



Nội dung buổi học

1. Giới thiệu mô hình MVC trong Laravel
2. Laravel Route
3. Laravel Blade





1. Giới thiệu mô hình MVC trong Laravel





Giới thiệu mô hình MVC trong Laravel

Mô hình MVC (Model-View-Controller) là một kiến trúc phần mềm phổ biến, được sử dụng rộng rãi trong phát triển ứng dụng web, đặc biệt là trong các framework như Laravel. Nó giúp tách biệt các thành phần của ứng dụng thành ba phần chính, giúp cho việc phát triển, bảo trì và mở rộng ứng dụng trở nên dễ dàng hơn.

3 thành phần chính của mô hình MVC:

- **Model:** Đại diện cho dữ liệu của ứng dụng, thường tương tác trực tiếp với cơ sở dữ liệu.
 - Truy xuất, cập nhật, xóa dữ liệu.
 - Thực hiện các quy tắc nghiệp vụ liên quan đến dữ liệu.
- **View:** Hiển thị giao diện người dùng dựa trên dữ liệu từ Model.
 - Nhận dữ liệu từ Controller và hiển thị ra màn hình.
 - Không chứa bất kỳ logic nghiệp vụ nào.
- **Controller:** Nhận các yêu cầu từ người dùng, xử lý các yêu cầu đó và trả về kết quả cho View.
 - Nhận dữ liệu từ View.
 - Gọi các phương thức trong Model để xử lý dữ liệu.
 - Chọn View phù hợp để hiển thị kết quả.

Xem ví dụ code.



2. Laravel Route



Route trong Laravel

Trong Laravel, routing (định tuyến) là một tính năng quan trọng giúp bạn định nghĩa các URL và hành vi của chúng trong ứng dụng web. Routing trong Laravel rất linh hoạt và mạnh mẽ, cho phép bạn dễ dàng quản lý các đường dẫn và ánh xạ chúng tới các controller hoặc hành động cụ thể. Nói cách khác, khi người dùng nhập một URL vào trình duyệt, Laravel sẽ tìm kiếm route phù hợp và thực thi đoạn code tương ứng.

Tìm hiểu về Routing trong Laravel trên trang chủ: <https://laravel.com/docs/11.x/routing>

Cấu trúc cơ bản của một Route:

- `Route::get()`: Định nghĩa một route nhận yêu cầu GET.
- `'/home'`: URL mà route này sẽ xử lý.
- `function()`: Một hàm vô danh chứa logic xử lý yêu cầu.

```
Route::get('/home', function () {  
    return 'Xin chào, bạn đang ở trang chủ!';  
});
```

Xem ví dụ code.





Các loại Route

Các loại route trong Laravel

- **Route GET:** Xử lý các yêu cầu **GET** (để lấy dữ liệu).
- **Route POST:** Xử lý các yêu cầu **POST** (để gửi dữ liệu).
- **Route PUT:** Xử lý các yêu cầu **PUT** (để cập nhật dữ liệu).
- **Route DELETE:** Xử lý các yêu cầu **DELETE** (để xóa dữ liệu).
- **Route ANY:** Xử lý tất cả các loại yêu cầu.

Xem ví dụ code.

Các tính năng với route trong Laravel

- **Route parameters:** Bạn có thể định nghĩa các tham số trong route để làm cho route trở nên linh hoạt hơn.
- **Route groups:** Bạn có thể nhóm các route lại để dễ quản lý và áp dụng các **middleware** chung.
- **Route name:** Bạn có thể đặt tên cho route để dễ dàng tham chiếu trong các view hoặc khi định nghĩa redirect..
- **Route model binding:** Laravel hỗ trợ tự động tiêm model vào controller dựa trên route parameter.

Xem ví dụ code.



3. Laravel Blade



Blade template trong Laravel

Laravel Blade là một template engine đơn giản nhưng mạnh mẽ, được tích hợp sẵn trong Laravel. Nó cung cấp một cú pháp trực quan và dễ sử dụng để tạo các view (giao diện người dùng) trong ứng dụng Laravel. Blade không hạn chế bạn sử dụng PHP thuần trong view, nhưng nó cung cấp một số cú pháp đặc biệt để làm cho việc tạo view trở nên hiệu quả hơn.

Tìm hiểu về Routing trong Laravel trên trang chủ:

<https://laravel.com/docs/11.x/views>

<https://laravel.com/docs/11.x/blade>

Cách gửi dữ liệu từ **Controller** tới **view** đang sử dụng **blade template**:

- Sử dụng hàm `with()`
- Sử dụng `compact()`
- Truyền dữ liệu trực tiếp vào view

Xem ví dụ code.





Blade template trong Laravel

Hiển thị dữ liệu:

- Dấu ngoặc nhọn kép: Sử dụng để hiển thị giá trị của một biến
- @echo: Cũng dùng để hiển thị dữ liệu
- Nếu bạn muốn hiển thị nội dung không escape, bạn có thể sử dụng cú pháp {!! !!}

Câu điều kiện:

- @if, @else, @elseif: Sử dụng để kiểm tra điều kiện

```
@if (count($users) > 0)
    <ul>
        @foreach ($users as $user)
            <li>{{ $user->name }}</li>
        @endforeach
    </ul>
@else
    <p>Không có người dùng nào.</p>
@endif
```

Xem ví dụ code.





Blade template trong Laravel

Vòng lặp:

- @for: Lặp một số lần xác định
- @foreach: Lặp qua một mảng hoặc đối tượng
- @forelse: Sử dụng @forelse để lặp qua một tập hợp và hiển thị một thông báo nếu tập hợp rỗng
- @while: Vòng lặp While

```
@for ($i = 0; $i < 10; $i++)  
    <p>Lặp lần thứ {{ $i }}</p>  
@endfor  
  
@foreach ($posts as $post)  
    <div>  
        <h2>{{ $post->title }}</h2>  
        <p>{{ $post->content }}</p>  
    </div>  
@endforeach
```

```
@forelse ($users as $user)  
    <p>Người dùng: {{ $user->name }}</p>  
@empty  
    <p>Không có người dùng nào.</p>  
@endforelse  
  
@while (true)  
    <p>Vòng lặp vô hạn.</p>  
@endwhile
```

Xem ví dụ code.





Blade template trong Laravel

Chỉ thị:

- @include: Nhúng một view khác vào view hiện tại
- @yield: Định nghĩa một section để các view con có thể thay thế
- @section, @endsection: Định nghĩa một section để các view cha có thể thay thế hoặc mở rộng
- @csrf: Bảo vệ chống tấn công CSRF
- @auth và @guest: Kiểm tra nếu người dùng đã xác thực hoặc là khách

```
@include('header')
<p>Nội dung chính của trang.</p>
@include('footer')

<div class="container">
    @yield('content')
</div>

@extends('layouts.app')
@section('title', 'Trang Chủ')
@section('content')
    <h1>Chào mừng đến với trang chủ</h1>
@endsection
```

```
<form method="POST" action="/profile">
    @csrf
    <!-- Form fields -->
</form>

@auth
    <p>Chào, {{ auth()->user()->name }}!</p>
@endauth

@guest
    <p>Vui lòng đăng nhập.</p>
@endguest
```

Xem ví dụ code.



