



Bài 14

Routing trong Laravel

Module: BOOTCAMP WEB-BACKEND DEVELOPMENT

- Trình bày được cơ chế routing trong Laravel
- Tạo được các route trong Laravel
- Trình bày được convention đối với URL của ứng dụng Web
- Tạo được route có tham số
- Sử dụng được named route
- Sử dụng được route group
- Định nghĩa được URI với các phương thức khác nhau như: GET, POST, PUT, PATCH, DELETE

Routing là gì

- Routing là cơ chế định tuyến
- Xác định cách hành xử đối với một đường dẫn
- Route gồm 2 phần
 - Đường dẫn URI : địa chỉ người dùng nhập vào
 - **Closure** - điểm đến: hàm để chỉ ra cách xử lý

```
Route::get('foo', function () {  
    return 'Hello World';  
});
```

File Route mặc định

- Tất cả các route được định nghĩa ở trong file route, ở trong thư mục routes, được tự động tải bởi framework.
- **routes/web.php** định nghĩa route cho giao diện web
 - Được gán vào thuộc nhóm middleware web, cung cấp một số tính năng như **session** và bảo mật **CSRF**
- **routes/api.php** được gán vào nhóm middleware **api**

Các phương thức Router



Laravel cho phép bạn đăng ký routes với nhiều phương thức HTTP

```
Route::get($uri, $callback);  
Route::post($uri, $callback);  
Route::put($uri, $callback);  
Route::patch($uri, $callback);  
Route::delete($uri, $callback);  
Route::options($uri, $callback);
```

```
Route::match(['get', 'post'], '/', function () {  
    //  
});  
  
Route::any('foo', function () {  
    //  
});
```

Bảo mật CSRF



- **CSRF (Cross Site Request Forgery)** là kĩ thuật tấn công bằng cách sử dụng quyền chứng thực của người sử dụng đối với 1 website khác
- Tất cả các HTML form có method là **POST**, **PUT** hoặc **DELETE** đều chỉ đến routes được định nghĩa trong middleware **web** thì cần được thêm trường **CSRF token**

Tham số route bắt buộc



- Tham số bắt buộc được đặt trong dấu {}

```
Route::get('user/{id}', function ($id) {  
    return 'User '.$id;  
});
```

- Định nghĩa nhiều tham số theo yêu cầu

```
Route::get('posts/{post}/comments/{comment}', function ($postId,  
    //  
});
```

Tham số route bắt buộc



- Tham số route luôn luôn phải nằm trong **dấu ngoặc nhọn {}**
- Các tham số route không được phép chứa ký tự **dấu gạch ngang -**
- Bạn có thể dùng ký tự **dấu gạch chân _** để thay thế

Tham số route tùy chọn

- Tham số tùy chọn là tham số có thể được truyền vào hoặc không
- Sử dụng dấu ? sau tên tham số để quy định nó là tham số tùy chọn
- Nên gán giá trị mặc định cho tham số tùy chọn

Tên routes

- Tên routes cho phép đặt tên thuận tiện cho các URL hoặc chuyển hướng cho các route cụ thể.
- Có thể chỉ định một tên cho route bằng cách thêm **name** vào định nghĩa route

```
Route::get('user/profile', function () {  
    //  
})->name('profile');
```

- Có thể chỉ định tên route cho controller

```
Route::get('user/profile', 'UserController@showProfile')->name('profile');
```

Tạo URL từ route đã đặt tên

- Sử dụng hàm `route()` để tạo ra URL từ tên của một route đã định nghĩa trước đó

```
// Generating URLs...  
$url = route('profile');  
  
// Generating Redirects...  
return redirect()->route('profile');
```

Tạo URL từ route đã đặt tên

- Trong trường hợp route có tham số, truyền danh sách các tham số thông qua một mảng:

```
Route::get('user/{id}/profile', function ($id) {  
    //  
})->name('profile');  
  
$url = route('profile', ['id' => 1]);
```

Nhóm Route

- Các route có liên quan có thể được nhóm lại với nhau, để chia sẻ chung các cấu hình, các cơ chế bảo mật, các cơ chế filter...
- Chẳng hạn, có các nhóm route khác nhau đối với từng chức năng khác nhau, hoặc đối với từng người dùng khác nhau
- Ví dụ: nhóm route dành cho admin, nhóm route dành cho guest, nhóm route dành cho người dùng đã đăng nhập...

Nhóm Route - Middleware



- Sử dụng hàm **middleware** để gán middleware cho tất cả các route trong một nhóm
- Middleware sẽ được thực hiện theo thứ tự được gọi

```
Route::group(['middleware' => 'auth'], function () {  
    Route::get('/', function () {  
        // Uses Auth Middleware  
    });  
  
    Route::get('user/profile', function () {  
        // Uses Auth Middleware  
    });  
});
```

Nhóm Route – Namespace

- Một trường hợp sử dụng chung cho nhóm route giống như PHP **namespace** được chỉ định với một nhóm của controllers.
- Sử dụng tham số namespace trong mảng thuộc tính

```
Route::group(['namespace' => 'Admin'], function() {  
    // Controllers Within The "App\Http\Controllers\Admin" Namespace  
});
```

Nhóm Route – Tên miền con

- Nhóm route có thể được sử dụng để xử lý các routing tên miền con.
- Tên miền con được gán tham số route như URIs, cho phép lấy một phần của tên miền con để sử dụng bên trong route hoặc controller.
- Tên miền con được xác định bằng cách sử dụng từ khóa domain

```
Route::group(['domain' => '{account}.myapp.com'], function () {  
    Route::get('user/{id}', function ($account, $id) {  
        //  
    });  
});
```


Nhóm Route – Tiền tố Route

- Thuộc tính **prefix** có thể sử dụng để thêm tiền tố cho mỗi route trong một nhóm với một URL.
- Ví dụ, có thể gán tiền tố **admin** cho tất cả các route URLs trong nhóm

```
Route::group(['prefix' => 'admin'], function () {  
    Route::get('users', function () {  
        // Matches The "/admin/users" URL  
    });  
});
```

Form Method Spoofing



- HTML forms không hỗ trợ các method **PUT, PATCH, DELETE**
- Khi định nghĩa **PUT, PATCH** hoặc **DELETE** cần phải thêm một trường hidden **_method** vào form.
- Giá trị của trường hidden là **_method** sẽ sử dụng phương thức HTTP request:

```
<form action="/foo/bar" method="POST">  
  <input type="hidden" name="_method" value="PUT">  
  <input type="hidden" name="_token" value="KqE0LAongzv9bIP6a0171EapbUYQDzjbwAwj0wJ0">  
</form>
```

Truy cập route hiện tại

- Sử dụng phương thức `current` hoặc `currentRouteName`, hoặc `currentRouteAction` trên `Route` facade để truy cập thông tin về request route hiện tại

```
$route = Route::current();
```

```
$name = Route::currentRouteName();
```

```
$action = Route::currentRouteAction();
```

Tóm tắt bài học



Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: ***Laravel Controller***