



Bài 15

Laravel Controller

Module: BOOTCAMP WEB-BACKEND DEVELOPMENT

Mục tiêu



- Trình bày được ý nghĩa của controller
- Tạo được controller cơ bản
- Thao tác được với form trong ứng dụng Laravel
- Trình bày được kiến trúc Repository
- Triển khai được kiến trúc Repository

Cơ bản về Controllers

Định nghĩa Controllers

Controllers và Namespaces

Một Action Controllers

- Controller đảm nhận xử lý các request và trả về response
- Các controller được đặt trong *app/Http/Controllers*
- Controller chứa các action
- Các action của controller nhận được request thông qua việc điều hướng của route

Định nghĩa Controllers



- Thừa kế lớp Controller

```
<?php

namespace App\Http\Controllers;

use App\User;
use App\Http\Controllers\Controller;

class UserController extends Controller
{
    /**
     * Show the profile for the given user.
     *
     * @param int $id
     * @return Response
     */
    public function show($id)
    {
        return view('user.profile', ['user' => User::findOrFail($id)]
    }
}
```

- Định nghĩa một route cho action của controller

```
Route::get('user/{id}', 'UserController@show');
```



Controllers không **yêu cầu** kế thừa từ base class. Tuy nhiên, bạn sẽ không có thêm một số tính năng như một số phương thức `middleware`, `validate`, và `dispatch`.

Controllers & Namespaces



- Chỉ định tên lớp sau `App\Http\Controllers` namespace

```
Route::get('foo', 'Photos\AdminController@method');
```

Controller với chỉ một action

- Nếu bạn muốn định nghĩa một controller xử lý duy nhất một action, bạn có thể dùng phương thức `__invoke` trong controller
- Đăng ký một route cho một action controllers, không cần xác định phương thức

```
Route::get('user/{id}', 'ShowProfile');
```

```
<?php

namespace App\Http\Controllers;

use App\User;
use App\Http\Controllers\Controller;

class ShowProfile extends Controller
{
    /**
     * Show the profile for the given user.
     *
     * @param int $id
     * @return Response
     */
    public function __invoke($id)
    {
        return view('user.profile', ['user' => User::findOrFail($id)]
    }
}
```


Controller Middleware

Controller là gì

Controller dùng thế nào

Controller Middleware



- **Middleware** được gán cho controller route ở trong file route

```
Route::get('profile', 'UserController@show')->middleware('auth');
```

Controller Middleware



- Có thể gán middleware trong constructor của controller.
- Có thể gán middleware sử dụng Closure.

```
$this->middleware(function ($request, $next) {  
    // ...  
  
    return $next($request);  
});
```

```
class UserController extends Controller  
{  
    /**  
     * Instantiate a new new controller instance.  
     *  
     * @return void  
     */  
    public function __construct()  
    {  
        $this->middleware('auth');  
  
        $this->middleware('log')->only('index');  
  
        $this->middleware('subscribed')->except('store');  
    }  
}
```



Bạn có thể gán middleware cho một tập con các action của controller; tuy nhiên, tập con action có thể to ra khi controller của bạn nhiều action. Vì thế, nên cân nhắc việc chia thành nhiều controller nhỏ hơn.

Resource Controllers

Từng phần Resource Routes

Tên Resource Routes

Tên tham số Resource Route

Bổ sung Resource Controllers

Resource Controllers



- Có thể tạo ra Controller với các route gắn với các phương thức CRUD bằng cách thêm tham số --resource khi tạo controller

```
php artisan make:controller PhotoController --resource
```

- Đăng ký một resourceful route cho controller:

```
Route::resource('photos', 'PhotoController');
```

Các action của Resource Controller



Verb	URI	Action	Route Name
GET	/photos	index	photos.index
GET	/photos/create	create	photos.create
POST	/photos	store	photos.store
GET	/photos/{photo}	show	photos.show
GET	/photos/{photo}/edit	edit	photos.edit
PUT/PATCH	/photos/{photo}	update	photos.update
DELETE	/photos/{photo}	destroy	photos.destroy

Đặt tên cho Resource Routes

- Mặc định, tất cả các action của resource controller đều có tên route
- Có thể ghi đè tên mặc định bằng cách truyền thêm mảng chứa **names** với tùy chọn:

```
Route::resource('photo', 'PhotoController', ['names' => [  
    'create' => 'photo.build'  
]]);
```


Tên tham số Resource Route

- Mặc định, `Route::resource` sinh ra tham số route cho resource routes dựa trên tên của resource
- Có thể ghi đè cho từng phần resource cơ bản bằng cách truyền `parameters` trong mảng

```
Route::resource('user', 'AdminController', ['parameters' => [  
    'user' => 'admin_user'  
]]);
```

Ví dụ trên sẽ tạo ra những URI sau cho route `show` của resource:

```
/user/{admin_user}
```

Constructor Injection



- [Service container](#) của Laravel chịu trách nhiệm xử lý tất cả các controllers
- Các dependency sẽ tự động xử lý và tiêm vào (inject) trong controller

```
<?php

namespace App\Http\Controllers;

use App\Repositories\UserRepository;

class UserController extends Controller
{
    /**
     * The user repository instance.
     */
    protected $users;

    /**
     * Create a new controller instance.
     *
     * @param UserRepository $users
     * @return void
     */
    public function __construct(UserRepository $users)
    {
        $this->users = $users;
    }
}
```

Tóm tắt bài học



Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: ***Views & Blade***