

# Laravel

Ths. Ngô Lê Quân  
[ngolequanit@gmail.com](mailto:ngolequanit@gmail.com)  
[www.Vku.udn.vn](http://www.Vku.udn.vn)

## Bài 8. Validation form

### ❑ 8.1. Validation form là gì?

- Một tính năng khác khá hay, và được dùng thường xuyên của Laravel, đó là validation các giá trị nhập từ form.
- Dùng kiểm tra các giá trị nhập từ các thành phần của form có đúng hay không, có phù hợp với các yêu cầu được khai báo hay không.
- Các giá trị validation dễ điều khiển, đa dạng, có thể thêm các thuộc tính khác nếu cần.
- Kết quả sẽ trả về lỗi nếu không đạt yêu cầu validation, nội dung lỗi có thể tùy biến thay đổi dễ dàng

## Bài 8. Validation form

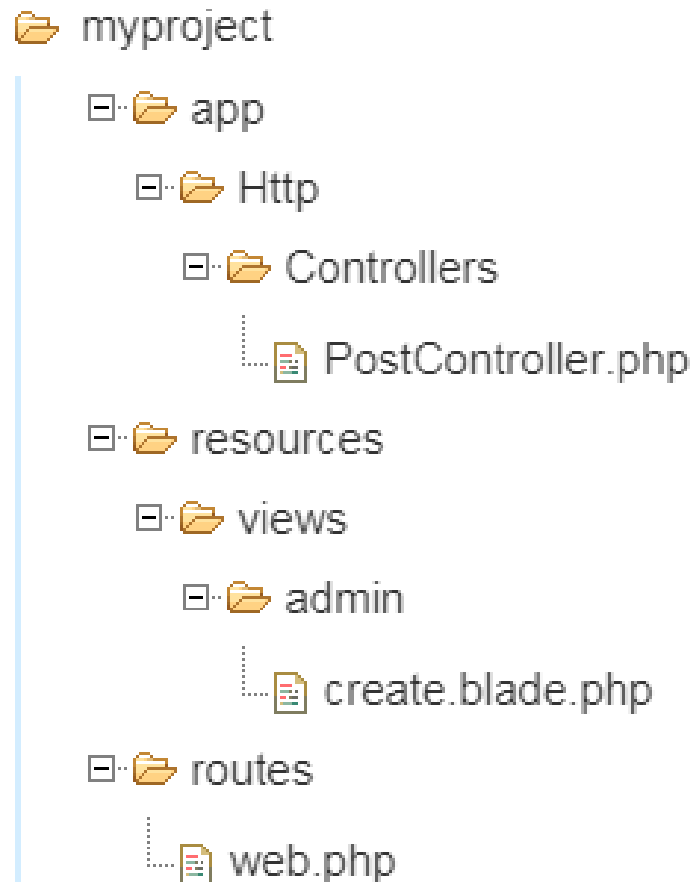
### ❑ 8.2. Thực hiện một validation form

- Bước 1 **TẠO TRANG VIEW**: **create.blade.php** có nội dung là một form với vài nội dung nhập dữ liệu.
- Bước 2. tạo Controller điều khiển các giá trị validation.
- Bước 3. Tạo một Routing điều hướng trang hiển thị nhập và trang hiển thị dữ liệu lỗi hay thông báo thành công sau khi form được nhấn Submit.

## Bài 8. Validation form

### ❑ 8.2. Thực hiện một validation form

Tạo nội dung các file trên theo cấu trúc thư mục sau:



## Bài 8. Validation form

### 8.2. Thực hiện một validation form

**TẠO TRANG VIEW:** create.blade.php

Đặt file này trong thư mục */resources/views/admin/*, với nội dung sau:

```
<!doctype html>
<html>
<head>
  <title>Login Form</title>
</head>

<body>
  <form method="post" action="/admin/create">
    @csrf
    <p>Title<br>
      <input type="text" name="title" value=""></p>

    <p>Description<br>
      <textarea rows="5" cols="40" name="description"></textarea></p>

    <p><button type="submit">Submit</button></p>
  </form>
</body>
</html>
```

## Bài 8. Validation form

### 8.2. Thực hiện một validation form

- `method="post"` - phương thức post của form.
- `@csrf` - tạo một chuỗi CSRF ngẫu nhiên, thuộc tính cần được sử dụng thường cho form, đây là một kỹ thuật tấn công sử dụng quyền chứng thực của người dùng để thực hiện các hành động trên một website khác.  
`action="/admin/create"` - khi click button Submit, sẽ post dữ liệu tới */admin/create*, cũng là chính trang View này.
- `name="title"` - dữ liệu sẽ được nhận thông qua name là title.
- `name="description"` - dữ liệu sẽ được nhận thông qua name là description.
- `type="submit"` - Dữ liệu sẽ gửi thông qua hành động click button.

## Bài 8. Validation form

### 8.2. Thực hiện một validation form

#### TẠO MỘT CONTROLLER: PostController.php

Tạo Controller bằng lệnh Artisan:

**php artisan make:controller PostController --resource**

Controller *PostController.php* này được đặt trong thư mục */app/Http/Controllers/*

```
>php artisan make:controller PostController --resource
```

## Bài 8. Validation form

### 8.2. Thực hiện một validation form

Thêm nội dung cho Controller này với function create()

- hiển thị trang xem nội dung form, store() thiết lập các request

```
public function create()  
{  
    return view('admin.create');  
}
```

```
public function store(Request $request)  
{  
    $this->validate($request, [  
        'title' => 'required',  
        'description' => 'required',  
    ]);  
}
```



## Bài 8. Validation form

### 8.2. Thực hiện một validation form

- TẠO MỘT ROUTING điều hướng nội dung  
function showform() trong Controller vừa tạo

**Route::get('/admin/create', 'PostController@create');**  
**Route::post('/admin/create', 'PostController@store');**

```
Route::get('/admin/create', 'PostController@create');  
Route::post('/admin/create', 'PostController@store');
```

## Bài 8. Validation form

### 8.3. Cú pháp viết một validation

#### Cú pháp cơ bản

'name một thẻ của\_form'=>'property1|property2|...'

Ví dụ

```
public function validationform(Request $request) { $this->validation($request,[  
'title'=>'required|string',  
'tel'=>'required|numeric' ]); }
```

## Bài 8. Validation form

### 8.3. Cú pháp viết một validation

**Cú pháp với giá trị**

**'name một thẻ của\_form' => 'property1:value1|property2:value2|...'**

Ví dụ

```
public function validationform(Request $request) { $this->validation($request,[  
'code'=>'digits:3',  
'age'=>'between:18,30'  
]); }
```

## Bài 8. Validation form

### 8.4. Validation thường dùng

Rule	Mô tả	VD
between:min,max ax	Dữ liệu nhập phải nằm trong giá trị min và max, giá trị có thể là chuỗi, số và file.	'age' => 'between:18,30'
confirmed	Dữ liệu nhập phải trùng khớp với name_confirmation	VD có 2 field: <i>name="password"</i> và <i>name="password_confirmation"</i> thì giá trị Validation sẽ là: 'password' =>'required confirmed min:8'
date	Dữ liệu nhập phải là ngày tháng, phù hợp với hàm strtotime của PHP.	'date' => 'date'
alpha	Dữ liệu nhập phải là chữ.	'name' => 'alpha'

## Bài 8. Validation form

### 8.4. Validation thường dùng

Rule	Mô tả	VD
<code>date_format:format</code>	Dữ liệu nhập phải giống định dạng với <i>format</i> , định dạng phải phù hợp với hàm <code>date_parse_from_format</code> của PHP.	'date' => 'date_format:d/m/Y'
<code>email</code>	Dữ liệu nhập phải là địa chỉ email.	'email' => 'email'
<code>size:value</code>	Dữ liệu nhập phải cùng kiểu với <i>value</i> , chuỗi thì là string, số thì là integer,...	'file' => 'size:300'
<code>starts_with:foo, bar,...</code>	Dữ liệu nhập phải bắt đầu một giá trị đưa trước.	'number' => 'starts_with:5'

## Bài 8. Validation form

### 8.5. Tùy chỉnh nội dung hiển thị lỗi

- Thông báo lỗi hiển thị là dạng mặc định có trong Laravel, trong phần này, chúng ta sẽ tùy chỉnh câu hiển thị lỗi như mong muốn, ví dụ hiển thị bằng tiếng Việt.
- Dựa theo kết quả của bài `create.blade.php` đã làm, chúng ta thêm vào controller `PostController.php` một biến *messages* với nội dung như sau:

## Bài 8. Validation form

### 8.5. Tùy chỉnh nội dung hiển thị lỗi

#### ➤ File Controller: PostController.php

```
public function store(Request $request)
{
    $messages = [
        'title.required' => 'Tiêu đề bắt buộc nhập',
        'title.max' => 'Tiêu đề không được vượt quá 255 ký tự',
        'description.required' => 'Nội dung mô tả bắt buộc nhập'
    ];

    $this->validate($request, [
        'title' => 'required|max:255',
        'description' => 'required',
    ], $messages);
}
```

## Bài 8. Validation form

### 8.5. Tùy chỉnh nội dung hiển thị lỗi

#### ➤ Kết quả

- Tiêu đề bắt buộc nhập
- Nội dung mô tả bắt buộc nhập

Title

Description

Submit



## Bài 8. Validation form

### 8.6. Tùy chỉnh vị trí hiển thị lỗi

#### ➤ Kết quả

Title

Tiêu đề bắt buộc nhập

Description

Nội dung mô tả bắt buộc nhập

Submit

## Bài 8. Validation form

### 8.6. Tùy chỉnh vị trí hiển thị lỗi

Laravel cung cấp một cách tùy chỉnh khá hay giúp điều khiển vị trí xuất hiện lỗi khá thuận tiện, đó là cách sử dụng hàm có sẵn `$errors->first('name')`.

Để thực hiện việc này, chúng ta cần viết lại controller *PostController.php* như sau:

```
$this->validate($request, [
    'title' => 'required|max:255',
    'description' => 'required',
], $messages);

$errors = $validate->errors();
```

## Bài 8. Validation form

### 8.6. Tùy chỉnh vị trí hiển thị lỗi

#### VIẾT LẠI THÔNG BÁO LỖI TẠI VIEW: create.blade.php

```
<p>Title<br>
    <input type="text" name="title" value=""><br>
    <span class="error-message">{{ $errors->first('title') }}</span></p>

<p>Description<br>
    <textarea rows="5" cols="40" name="description"></textarea><br>
    <span class="error-message">{{ $errors->first('description') }}</span></p>
```