

Validation Laravel

Tổng quan về Validation Laravel

- Xác thực dữ liệu (Validate) là công việc không thể thiếu trong lập trình khi nhận dữ liệu từ phía client
- Laravel cung cấp cho chúng ta rất nhiều phương thức để xác thực dữ liệu
- Ngoài việc hỗ trợ rất mạnh việc sử dụng Validate, Laravel còn hỗ trợ rất mạnh việc phát triển, mở rộng
- Laravel có 3 cách Validate:
 - Sử dụng phương thức validate() từ lớp Request()
 - Sử dụng Form Request
 - Sử dụng lớp Validator()

Validation bằng phương thức validate()

Với phương thức này, bạn cần khởi tạo Request() trước, sau đó sử dụng theo cú pháp sau:

```
$request->validate($rules, $messages);
```

- **\$rules** là một mảng dữ liệu với key là các input name, value là các rule mà bạn muốn ràng buộc.
- **\$messages** là một mảng dữ liệu chứa các nội dung thông báo. Nếu không điền trường này sẽ lấy thông báo mặc định

Xem qua ví dụ sau để hiểu rõ hơn

```
$request->validate([  
    'age' => 'required|integer|min:30'  
]);
```

Các bạn thấy các rule được ngăn cách với nhau bởi dấu gạch |. Hoặc bạn có thể chuyển thành mảng như sau:

```
$request->validate([  
    'age' => ['required', 'integer', 'min:30']  
]);
```

Nếu validate không thành công, Laravel sẽ tự động redirect về request trước kèm theo các thông báo được gán vào Flash Session

Nếu cần trả về các lỗi kiểu JSON, bạn cần thêm header

```
Content-Type: application/json
```

Validation sử dụng Form Request

Laravel cung cấp cho lập trình viên cách tạo ra các lớp Validate riêng để dễ dàng tái sử dụng, quản lý và nâng cấp. Lớp riêng này sẽ kế thừa từ lớp `FormRequest()`

Để tạo Form Request, bạn sử dụng câu lệnh sau

```
php artisan make:request TenRequest
```

Tên request bạn tự đặt theo ý muốn và hãy đặt tên quy tắc **PascalCase**

Xem ví dụ sau để hiểu rõ hơn

Tạo request có tên **ProductRequest** bằng cách sử dụng câu lệnh:

```
php artisan make:request ProductRequest
```

Lúc này, file `ProductRequest.php` sẽ được tạo ra trong thư mục:

```
app/Http/Requests/ProductRequest.php
```

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class ProductRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this
    request.
     *
     * @return bool
     */
    public function authorize()
    {
        return false;
    }

    /**
     * Get the validation rules that apply to the
    request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            //
        ];
    }
}

```

- Phương thức **authorize()** sẽ cho phép người dùng thực thi request này hay không (true = cho phép, false = không cho phép)

- Phương thức **rules()** sẽ chứa các rules cần validate

Nếu muốn thay đổi nội dung thông báo, hãy bổ sung phương thức `messages()` với 2 cách khai báo sau:

```
public function messages()
{
    return ['field.rule' => 'Nội dung thông báo'];
}
```

```
public function messages()
{
    return ['rule' => ':attribute nội dung thông báo'];
}
```

:attribute sẽ tự động lấy tên trường

Nếu muốn thay đổi tên trường, hãy bổ sung phương thức `attributes()` với cú pháp khai báo như sau:

```
public function attributes()
{
    return [
        'field' => 'Tên field mới',
    ];
}
```

Validation sử dụng lớp Validator

Laravel cung cấp sẵn class `Validator`, bạn có thể sử dụng class này để Validate dữ liệu.

Cú pháp:

```
Validator::make($inputs, $rules, $messages, $attributes);
```

- `$inputs` là một mảng dữ liệu chứa các dữ liệu bạn cần validate (Thường truyền bằng: `$request->all()`)
- `$rules` là một mảng dữ liệu với key là các input name, value là các rule mà bạn muốn ràng buộc
- `$messages` là một mảng dữ liệu chứa các nội dung thông báo của bạn. Trường này có thể bỏ trống
- `$attributes` là một mảng dữ liệu chứa các tên trường của bạn. Trường này có thể bỏ trống.

Phương thức này sẽ trả về 1 object. Nếu muốn thực hiện validate, bạn cần gọi phương thức `validate()`

Ngoài ra, bạn có thể kiểm tra xem validate thành công hay không bằng cách sử dụng phương thức `fails()`

Cùng xem 2 ví dụ sau:

```
Validator::make($request->all(), [
    'name' => 'required|min:20',
    'content' => 'required',
])->validate();
```

```
$validate = Validator::make($request->all(), [
    'name' => 'required|min:20',
    'content' => 'required',
]);

if ($validate->fails()){
    //logic fails
}
```

Tạo thêm Rule Validation

Trong trường hợp các Rule của Laravel không đáp ứng được yêu cầu của dự án. Lúc này bạn tạo thêm Rule Validation theo các bước sau

Tạo Rule mới bằng lệnh sau

```
php artisan make:rule RuleName
```

Trong đó là **RuleName** là tên Rule Validation muốn tạo

Ví dụ muốn tạo Rule Uppercase

```
php artisan make:rule Uppercase
```

Lúc này, Rule sẽ được thêm vào folder `app/Rules`

Mở file: `app/Rules/Uppercase.php`

```
<?php

namespace App\Rules;

use Illuminate\Contracts\Validation\Rule;

class Uppercase implements Rule
{
    /**
     * Create a new rule instance.
     *
     * @return void
     */
    public function __construct()
    {
        //
    }

    /**
     * Determine if the validation rule passes.
     *
     */
}
```

```

    * @param string $attribute
    * @param mixed $value
    * @return bool
    */
    public function passes($attribute, $value)
    {
        //
    }

    /**
     * Get the validation error message.
     *
     * @return string
     */
    public function message()
    {
        return 'The validation error message.';
    }
}

```

- Phương thức **passes()** mà trả về true thì rule sẽ được pass và ngược lại.
- Phương thức **message()** sẽ trả về thông báo nếu như Rule không pass.

Sử dụng Rule Validation

Để sử dụng Rule Validation, hãy áp dụng theo cú pháp sau

```

use App\Rules\Uppercase;

$request->validate([
    'name' => ['required', 'string', new Uppercase],
]);

```

Cách sử dụng này, hoàn toàn áp dụng được với **Form Request**, **Validator**

Trong các trường hợp đơn giản, bạn có thể tạo thêm Rule bằng cách sử dụng Closure với cú pháp sau

```
function ($attribute, $value, $fail) {  
    if (condition) {  
        $fail(Message);  
    }  
}
```

- **\$attribute** là tên của trường cần validate.
- **\$value** là giá trị của trường cần validate.
- **\$fail** là một callback function sẽ được gọi khi validate fail.

Ví dụ:

```
$validator = Validator::make($request->all(), [  
    'name' => [  
        'required',  
        'min:20',  
        function ($attribute, $value, $fail) {  
            if ($value === 'admin') {  
                $fail('The '.$attribute.' is invalid.');            }  
        },  
    ],  
]);
```