**GIÁO TRÌNH LARAVEL - BUỔI 4**

**FORM REQUEST**

**HELPER VÀ XÂY DỰNG HÀM HIỂN THỊ LỖI Ô INPUT**

1. **From Request là gì ?**

*Form request giúp chúng ta tách biệt việc kiểm tra tính hợp lệ của ô input ra ngoài controller giúp code sạch và dễ nhìn hơn, dễ quản lý hơn.*

1. **Tạo và sử dụng From Request**

a.Tạo ra FromRequest

Cú pháp :

php artisan make:request nameFromRequest

Ví dụ :

php artisan make:request LoginRequest

b. Sử dụng lớp FormRequest trong Controllers

Để sử dụng lớp FormRequest trong Controllers thì trước hết các bạn phải gọi namespace của Request đó với cú pháp:

use App\Http\Requests\RequestName;

VD: Muốn gọi LoginRequest:

use App\Http\Requests\LoginRequest;

Tiếp đó hàm nào các bạn muốn sử dụng validation thì chỉ việc ánh xạ nó vào hàm với cú pháp:

public function nameFunction(RequestName $r)

{

//

}

VD:

public function postLogin(LoginRequest $request)

{

return $request->all();

}

c. Cấu hình Request

Bạn truy cập vào app/Http/Requests sẽ thấy có một file *LoginRequest.php*

Trong form request vừa tạo authorize chuyển từ false ->true:

|  |
| --- |
| public function authorize()  {  return true;  } |

\*Chú ý: Tất cả các file Request phải được đặt trong folder Requests

Rules và massages:

* Cấu hình chuỗi quy tắc các bạn phải viết trong hàm rules()
* Thay đổi nội dung báo lỗi viết trong massages()
* Nội dung cú pháp thì hoàn toàn giống phần validation

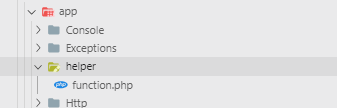
|  |
| --- |
| public function rules()  {  return [  //chứa các quy tắc  ];  }  public function messages()  {  return [  //chứa các thông báo lỗi tương ứng với từng ô input  ];  } |

1. **Helper và xây dựng hàm showErrors**

a. Helper và xây dựng hàm trong helper

*Helper là những hàm trợ giúp mình tự thêm hoặc có sẵn giúp hỗ trợ trong dự án mà bạn có thể gọi từ bất cứ nơi nào trong ứng dụng .*

Helper được để trong thư mục *app*. Vì vậy để tạo helper riêng ta tạo 1 thư mục mới có tên *helper* trong thư mục *app.*



Trong *helper* ta tạo 1 file tên *function.php* để chứa các hàm riêng

VD:

Một hàm đơn giản của function.php trong helper:

<?php

function hello(){

echo 'Xin chào các bạn';

}

b. Khai báo *helper* :

Để có thể sử dụng các hàm trong helper ta phải khai báo chúng :

* Vào *compser.json* khai báo vị trí file function.php trong phần autoload , ta khai báo vị trí file *function.php:*

"autoload": {

"psr-4": {

"App\\": "app/"

},

"classmap": [

"database/seeds",

"database/factories"

],

*"files": [*

*"app/helper/function.php"*

*]*

},

* Mở cửa sổ terminal của dự án ,chạy câu lệnh :

**composer dump-autoload**

Kiểm tra *autoload\_files.php* trong *vendor->composer* nếu có :

'4ef25952a3d3f7c399fe269e8e36b783' => $baseDir . '/app/helper/function.php',

Vậy là ta đã khai báo thành công và có thể sử dụng hàm xây dựng trong helper trong dự án .

VD:

Ta có thể sử dụng ở bên view để in ra câu chào

{{ hello() }}

c. Xây dựng hàm hiển thị lỗi ở view

Ta đưa đoạn code hiển thị lỗi ô input vào một function trong function.php và sửa về code php in ra các dòng lệnh hiển thị lỗi ô input với 1 biến thay thế tên ô input.

VD:

<?php

function showErrors($errors,$nameInput){

if ($errors->has($nameInput)) {

echo '<div class="alert alert-danger" role="alert">';

echo '<strong>'.$errors->first($name).'</strong>';

echo '</div>';

}

}