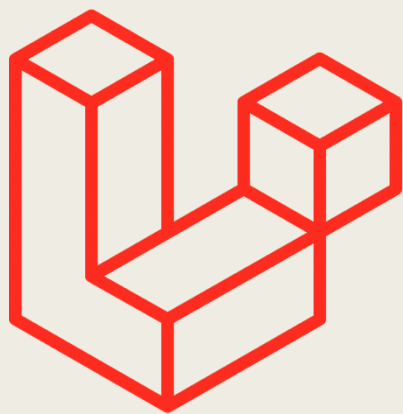


驗證

朱克剛



目的

- 對 Request 中的資料進行驗證，例如有些資料一定要有，有些資料長度有所限制...等
- Laravel 的驗證功能幫我們完成許多事情，可以少寫很多程式碼

表單驗證

- 要驗證表單中輸入的資料，雖然可以手動建立驗證，但建議使用 Laravel 幫我們產生的驗證類別，指令如下：
 - `php artisan make:request MyFormRequest`
- 產生的檔案位於 App\Http\Requests 目錄中

第 1 步：建立表單

- 檔名 form.blade.php

```
<form action="process" method="post">  
    @csrf  
    <input name="message"><p>  
    <input type="submit">  
</body>
```

第 2 步：產生 Controller

- 指令
 - `php artisan make:controller ProcessController`
- 內容為

```
use App\Http\Requests\MyFormRequest;  
  
class ProcessController extends Controller  
{  
    public function do(MyFormRequest $request) {  
        $request->validated();  
        return view('success');  
    }  
}
```

通過驗證後的網頁，請自行建立

第 3 步：MyFormRequest 內容

- 若希望表單中 message 格子內容最多五個字，並且一定要填的話

```
public function authorize()  
{  
    return true;  
}
```

改為 true，否則會出現網頁未授權訊息

```
public function rules()  
{  
    return [  
        'message' => 'required|max:5'  
    ];  
}
```

也可以這樣寫：['required', 'max:5']

第 4 步：設定路由

```
namespace App\Http\Controllers;
```

```
Route::post('/process', [ProcessController::class, 'do']);
```

第 5 步：錯誤處理

- 當驗證未過時，預設會將網頁退回到原本發出 http request 的網頁，也就是表單，在表單最上方加上：

```
@if ($errors->any())
    <div style="color:red;">
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif
```


第 6 步：執行看看

- 當 message 中為填入任何資料就送出，此時驗證不過，畫面如下



A screenshot of a web form validation error. At the top, a red bullet point followed by the text "The message field is required." is displayed. Below this message is an empty rectangular input field. At the bottom of the form area is a button with the text "送出查詢" (Submit Query).

- 預設錯誤訊息為英文，定義在 lang/en/validation.php 中

自訂錯誤訊息

- 在 MyFormRequest 中加上 message() 函數

```
public function messages() {  
    return [  
        'message.required' => '訊息欄:attribute未輸入資料',  
        'message.max' => '訊息欄:attribute輸入超過:max個字'  
    ];  
}
```

驗證不過保留原本輸入的資料

- 預設當驗證不過時，畫面會退回到原本的表單網頁，而原本輸入的資料會清除，若要保留原本輸入的資料（不然使用者會翻臉），表單修改如下：

```
<input name="message" value="{{ old('message') }}">
```

驗證不過時退回自訂頁面

- 預設為退回原先頁面，想要退回自訂頁面，在 MyFormRequest 中覆寫 \$redirect 變數即可，如下：
 - *protected \$redirect = '/invalidation';*

其他驗證規則

還有哪些驗證規則

- 多如牛毛的內建驗證規則

- <https://laravel.com/docs/9.x/validation#available-validation-rules>

密碼驗證

- 密碼輸入框通常有兩個，一個為密碼，一個確認用，命名規則為確認用的名稱要以「_confirmation」結尾，主名稱必須一致，如下

Password: `<input name="password"><p>`

Type Again: `<input name="password_confirmation">`

設定驗證規則

```
use Illuminate\Validation\Rules\Password;
```



記得加

```
public function rules()  
{  
    return [  
        'password' => ['required', 'confirmed', Password::min(8)]  
    ];  
}
```


內建的 Password 規則

- 檔案位於

- `vendor/laravel/framework/src/Illuminate/Validation/Rules/Password.php`

```
$rules = Password::min(8)
    ->mixedCase()      // 大小寫混合
    ->letters()         // 要有一個文字
    ->numbers()         // 要有一個數字
    ->symbols();        // 要有一個符號
```

唯一值驗證

- 此驗證必須與資料庫連線，因此需先設定 .env 檔
- 語法：'unique:table,column'
- 若希望表單中的 uid 欄位在資料庫中尚未出現，作法如下：

表單

```
<input name="uid">
```

驗證規則

```
public function rules()  
{  
    return [  
        'uid' => 'unique:UserInfo,uid'  
    ];  
}
```