

Laravel Day 2

Route與 Blade 模板變數傳遞

1-1 Route(路由)

**今天我們要學會讓網址對應到功能，
也就是網站的「路口交通指揮員」—— Route !**

基本路由

```
Route::get('/', function () {  
    return 'main page';  
});  
  
Route::get('/hello', function () {  
    return 'hello';  
});
```

- `Route::get()`：設定網址與回應的內容
- 第一個參數是網址，第二個是執行的函式

動態參數路由

```
Route::get('/hello/{name}', function ($name) {  
    return 'hello ' . $name . '!';  
});
```

- `{name}` 是路由參數
 - 網址變成 `/hello/Alice` 就會輸出 `hello Alice!`
-  適合用在會員頁、文章頁等「網址會變」的地方

重定向路由

```
Route::get('/hallo', function () {  
    return redirect('/hello');  
});
```

- `redirect()`：將使用者導向另一個網址
- 適合修錯字、替換舊網址

 實務常用於 SEO 或整理網站架構時

命名路由

```
Route::get('/welcome', function () { // 顯示歡迎畫面  
    return 'Welcome to Laravel!';  
})->name('welcome.page');
```

```
Route::get('/start', function () { // 導向到歡迎畫面  
    return redirect()->route('welcome.page');  
});
```

- `->name('xxx')` 給路由命名
- `redirect()->route('xxx')` 用命名導向

 實務上非常好用！以後就不怕網址改變

Fallback Route

```
Route::fallback(function () {  
    return 'Still got somewhere';  
});
```

- 網址如果對不到任何已定義的，就會跑這裡

 可以在這裡做自訂的 404 頁面

Artisan 工具：看所有路由

```
php artisan route:list
```

 可以看到所有定義好的路由，包括：

- 請求方法（GET / POST）
- 對應功能
- 路由名稱（有命名的話）

小結

每當有人打開一個網址，Laravel 的 Route 就會：

- 1 檢查這條網址對應的功能是什麼
- 2 決定回傳什麼資料或畫面
- 3 如果沒找到，還能跳出預設提示頁 (Fallback)

 學會 Route，就能做到這些事：

- 指定每個網址對應的功能
- 傳資料給畫面 (Blade)
- 實作簡單的轉址、錯誤處理

1-2 Blade模板變數傳遞

當使用者打開一個網址時，Laravel 會：

- 1 找到對應的 Route
- 2 Route 把資料準備好，交給 Blade 畫面
- 3 Blade 根據資料，產生 HTML 回傳給使用者

這就是從「網址」 → 「資料處理」 → 「畫面顯示」的完整流程！

Route 與 View 的關係

```
Route::get('/', function () {  
    return view('index', [  
        'name' => 'Celia'  
    ]);  
});
```

 `view()` 是 Laravel 把資料送到畫面的橋樑

- 第一個參數是 Blade 檔名
- 第二個是要傳給畫面的資料

資料怎麼傳？陣列 vs compact

```
// 方法一：傳陣列
return view('profile', [
  'name' => 'Ruru',
  'age' => 88
]);

// 方法二：用 compact
$name = 'Ruru';
$age = 88;
return view('profile', compact('name', 'age'));
```

- `compact('name', 'age')` 會幫你自動組成陣列

Blade 是什麼？

Blade 是 Laravel 專用的「畫面語法」，副檔名是 `.blade.php`

它幫你：

- ✓ 顯示變數（自動防止 XSS）
- ✓ 加上條件、迴圈、元件
- ✓ 寫 PHP 邏輯更簡潔

Blade：變數輸出

```
<!-- 最常見 -->
```

```
{{ $name }}
```

```
<!-- 不轉義 HTML -->
```

```
{!! $description !!}
```

- `{{ }}`：自動轉義 HTML，防止 XSS
- `{!! !!}`：不建議用除非你很確定資料安全！

Blade：條件語法

```
@if($age >= 18)
    <p>你是大人了</p>
@else
    <p>你還未成年</p>
@endif

@isset($name)
    <p>姓名是：{{ $name }}</p>
@endisset
```

- `@isset()` 比 `@if()` 更安全用於「變數有沒有存在」

Blade：迴圈語法

```
@foreach ($members as $member)  
    <li>{{ $member }}</li>  
@endforeach
```

 畫面中跑出每一個名字或清單項目

練習：

◆ 練習 1：命名路由與導向練習

- ! 建立 /welcome 路由，回傳文字 Welcome, Laravel!
 - ! 幫這條路由取一個名字（例如 welcome.page）
 - ! 再寫一條新的路由 /start，導向剛剛命名的頁面
-

◆ 練習 2：Blade 畫面資料傳遞練習

- ! 建立 /about 路由
- ! 傳入 team 與 members 資料
- ! 使用 Blade 模板畫面 about.blade.php 顯示資料清單

小結

✓ Route 是「負責接收網址請求」的守門員，你可以在這裡決定要回什麼頁面、送什麼資料

✓ Blade 是「幫你顯示資料的畫面語言」，用 `{{ $變數 }}` 就能把資料秀出來

✓ 想把後端資料放到畫面上，就是這樣做：
Route 傳 → Blade 接 → 使用者看見