

Laravel Day 3

任務清單練習 & 任務詳情頁

任務清單練習 (Task List)

前面我們學了：

- ✓ Route 怎麼傳資料到 Blade
- ✓ Blade 怎麼顯示變數和條件判斷

👉 今天，我們要讓這些技巧更有「實戰感」

想像一下：

你正在開發一個**任務管理小工具**，畫面上會顯示多個任務，每個任務有：

- 標題 | 說明 | 完成狀態

但...你還沒學到資料庫怎麼用 🤔

但我們可以先用 PHP 模擬資料 → 練習畫面邏輯！

從「一筆變數」 → 「一組清單」

從 `{{ $name }}` → `@foreach ($tasks as $task)`

這就是今天的主題！

1：建立 Task 類別（模擬資料）

```
// routes/web.php
class Task {
    public function __construct(
        public int $id,
        public string $title,
        public string $description,
        public ?string $long_description,
        public bool $completed,
        public string $created_at,
        public string $updated_at
    ) {}
}

// 模擬多筆任務
$tasks = [
    new Task(1, '任務A', '簡介...', null, true, '2025-06-01', '2025-06-01'),
    new Task(2, '任務B', '簡介...', null, false, '2025-06-02', '2025-06-02'),
    // ... 你可以自由新增
];
```

2：將這些假資料透過 Route 傳到 Blade 畫面

```
// routes/web.php

Route::get('/', function () use ($tasks) {
    return view('index', [
        'tasks' => $tasks
    ]);
})->name('tasks.index');
```

- `Route::get()`：設定首頁網址
- `use($tasks)`：引入外部變數
- `view('index', [...])`：傳資料給 Blade
- `name('tasks.index')`：命名路由，方便跳轉

3：建立 index.blade.php

```
<!-- resources/views/index.blade.php -->

@foreach ($tasks as $task)
    <div>
        <h2>{{ $task->title }}</h2>
        <p>{{ $task->description }}</p>
    </div>
@endforeach
<p>No tasks available.</p>
@empty
```

- `@foreach`：有資料就列出，沒資料顯示備註
- `{{ $task->title }}`：顯示任務標題；`{{ $task->description }}`：顯示任務描述
- `@empty`：如果沒任務，就顯示「No tasks available.」

4：啟動 Laravel 開發伺服器

```
php artisan serve
```

然後打開瀏覽器輸入：

<http://localhost:8000>

你應該會看到畫面列出一堆任務標題和描述 🎉

任務詳情頁

這次我們要完成什麼？

建立一個簡單的任務管理頁面，包含：

- 任務列表（`/tasks`）
- 任務詳情頁（`/tasks/{id}`）
- 頁面使用 Blade 模板、具備 layout 套版功能
- 以 PHP 類別＋陣列模擬資料庫內容

1. 建立 Task 類別（模擬資料）

route/web.php

```
class Task
{
    public function __construct(
        public int $id,
        public string $title,
        public string $description,
        public ?string $long_description,
        public bool $completed,
        public string $created_at,
        public string $updated_at
    ) {
    }
}
```

```
$tasks = [
    new Task(
        1,
        'Buy groceries',
        'Task 1 description',
        'Task 1 long description',
        false,
        '2023-03-01 12:00:00',
        '2023-03-01 12:00:00'
    ),
    new Task(
        2,
        'Sell old stuff',
        'Task 2 description',
        null,
        false,
        '2023-03-02 12:00:00',
        '2023-03-02 12:00:00'
    ),
    new Task(
        3,
        'Learn programming',
        'Task 3 description',
        'Task 3 long description',
        true,
        '2023-03-03 12:00:00',
        '2023-03-03 12:00:00'
    ),
    new Task(
        4,
        'Take dogs for a walk',
        'Task 4 description',
        null,
        false,
        '2023-03-04 12:00:00',
        '2023-03-04 12:00:00'
    ),
];
```

2. 建立 Route（網址與功能對應）

route/web.php 中加入：

```
Route::get('/', function () { // 首頁自動轉向 /tasks
    return redirect()->route('tasks.index');
});

Route::get('/tasks', function () use ($tasks) { // 任務清單頁
    return view('index', ['tasks' => $tasks]);
})->name('tasks.index');

Route::get('/tasks/{id}', function ($id) use ($tasks) { // 任務詳情頁（點進去看細節）
    $task = collect($tasks)->firstWhere('id', $id);
    if (!$task) {
        abort(404);
    }
    return view('show', ['task' => $task]);
})->name('tasks.show');
```

- collect()->firstWhere() 是從陣列中快速查找物件的方法

3. 建立共用樣板 `app.blade.php`

```
// resources/views/layouts/app.blade.php
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>Task List APP</title>
  </head>
  <body>
    <h1>@yield('title')</h1>
    <div>@yield('content')</div>
  </body>
</html>
```

- 用 @yield() 提供可替換區塊，讓頁面共用一致架構
- 每個子頁面都可以指定 @section() 來塞入內容

4：調整清單頁 `index.blade.php`

```
{{-- 引入模板 --}}
@extends('layouts.app')
@section('title', 'The list of the books')
@section('content')
    <div>
        @if (count($tasks))
            @foreach ($tasks as $task)
                <div>
                    <a href="{{ route('tasks.show', ['id' => $task->id]) }}">
                        {{ $task->title }}
                    </a>
                </div>
            @endforeach
        @else
            <div>There are no tasks</div>
        @endif
    </div>
@endsection
```

說明:

Blade 語法	用途描述
<code>@extends()</code>	指定要繼承哪個 layout 模板
<code>@section()</code>	開始定義要填進主模板的區塊內容

5. 建立詳情頁 `show.blade.php`

```
@extends('layouts.app')
@section('title', $task->title)

@section('content')
    <p>{{ $task->description }}</p>

    @if ($task->long_description)
        <p>{{ $task->long_description }}</p>
    @endif

    <p>{{ $task->created_at }}</p>
    <p>{{ $task->updated_at }}</p>
@endsection
```

- 可用 `@if` 檢查是否有long_description，再決定是否顯示

小結

- ✅ Task 類別讓我們能假裝有資料庫
- ✅ Route 負責準備資料 → Blade 將資料變畫面
- ✅ Blade 的 @if / @forelse 是清單呈現好幫手
- 📌 未來接上資料庫，只要換資料來源就能無痛轉換！

我們完成了清單頁（index）與詳情頁（show），搭配 Layout 模板建立更一致的畫面架構！