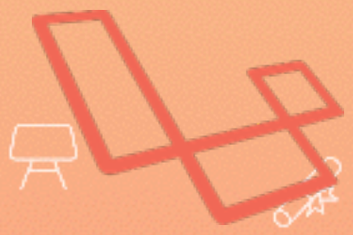
適用 5.1 版



CRUD綜合運用

范聖佑 Shengyou Fan

新北市樹林國小 (2015/07/09)

```php
/**
 * Display the specified resource.
 *
 * @param  int  $id
 * @return Response
 */
public function show($id)
{
    $post = Post::with('comments')->where('id', $id)->first();

    return View::make('post.show')->with('post', $post);
}
```

單元主題

- 了解何謂 CRUD？以及 CRUD 對應的動作
- 安裝 Form Builder 產生表單 DOM
- 了解更多 Eloquent 特殊功能
- 示範實作專案 CRUD 功能

CRUD 簡介

什麼是 CRUD？

- CRUD = 針對一個 resource 的標準四個操作
  - Create 建立資料
  - Read 讀取資料
  - Update 更新資料
  - Delete 刪除資料

RESTful 動作對應

- Laravel 的 RESTful Controler 動作對應依以下表格：

動詞	路徑	動作	名稱
GET	/resource	index	resource.index
GET	/resource/create	create	resource.create
POST	/resource	store	resource.store
GET	/resource/{id}	show	resource.show
GET	/resource/{id}/edit	edit	resource.edit
PUT/PATCH	/resource/{id}	update	resource.update
DELETE	/resource/{id}	destroy	resource.destroy

Route 動作設定

- 設定 Route 動作改對應至相應的 Controller

```
// app/Http/routes.php
Route::get(/* 略 */);
Route::post(/* 略 */);
Route::patch(/* 略 */);
Route::delete(/* 略 */);
```

- 也可以用 RESTful Resource Controllers，一行抵數行

```
// app/Http/routes.php
Route::resource('posts', 'PostsController');
```

使用 Form Builder

★ 參考文件：http://laravelcollective.com/docs/5.1/html

用 Form Builder 產生 DOM

- 直接在 View 裡寫 HTML DOM 是絕對沒有問題的
- 但若是可以用 Form Builder 來產生 HTML 的話，還可以搭配 Model 自動產生一些 DOM，省去自己寫 HTML 的時間
- Form Builder 在 Laravel 4 時是內建的，Laravel 5 以後從核心元件移出來，若想用的話要自己裝：

```
- "laravelcollective/html": "^5.1"
```

開啟/關閉表單

- 開啟表單

```
Form::open(['url' => '{foo/bar}', 'method' => 'POST'])
Form::open(['route' => '{route.name}', 'method' => 'POST'])
```

- Model 綁定

```
Form::model($post, ['route' => ['{route.name}', $post->id]])
```

- 關閉表單

```
Form::close()
```

產生表單元素

- 產生 Input、Label、Textarea

```
Form::label('{input name}', '{display text}')
Form::text('{field name}', '{value}', [{opts array}])
Form::email('{field name}', '{value}', [{opts array}])
Form::textarea('{field name}', '{value}', [{opts array}])
Form::radio('{field name}', '{value}', {default})
```

- 產生 Submit 按鈕

```
Form::submit('{field name}', [{opt_array}])
```

View 上面產生表單

- 在 View 上面產生需要使用的表單 HTML，並指定動作至對應的 Route

```
// resources/views/posts/create.blade.php
Form::open(['route' => 'posts.store', 'method' => 'POST'])
  /* 略 */
Form::close()


// resources/views/posts/edit.blade.php
// $post 是 Model
Form::model($post, ['route' => ['posts.update', $post->id],
'method' => 'PATCH'])
  /* 略 */
Form::close()
```

CRUD 綜合運用

寫入資料

- 在 Controller 裡接受表單送來的資料，並用 Model 把資料寫入資料庫

```
// app/Http/Controllers/PostsController.php
public function store(Request $request)
{
    $post = \App\Post::create($request->all());

     /* 略 */
}
```

更新資料

- 在 Controller 裡接受表單送來的資料，並用 Model 把資料更新至資料庫

```
// app/Http/Controllers/PostsController.php
public function update($id, Request $request)
{
    $post = \App\Post::find($id);

    $post->update($request->all());

     /* 略 */
}
```

刪除資料

- 在 Controller 裡接受表單送來的資料，並用 Model 把資料從資料庫裡刪除
- 利用 Model 的關聯設定，把關聯資料表內的資料也一起刪除

```
// app/Http/Controllers/PostsController.php
public function destroy($id)
{
    $post = \App\Post::find($id);

    foreach($post->comments as $comment) {
        $comment->delete();
    }

    $post->delete();

    /* 略 */
}
```

頁面跳轉

- 操作資料庫的動作完成後，依照需求跳轉至指定的頁面

```php
// app/Http/Controllers/PostsController.php
public function store(Request $request)
{
    /* 略 */

    return redirect()->route('posts.show', $post->id);
}

// app/Http/Controllers/PostsController.php
public function destroy($id)
{
    /* 略 */

    return redirect()->route('posts.index');
}
```

自動分頁

- Laravel 的 Model 在取筆數時，可以自動幫我們做分頁，在頁面上也可以自動幫我們產生 Bootstrap 的 Pagination 元件

```
// app/Http/Controllers/PostsController.php
public function index()
{
    $posts = \App\Post::orderBy('created_at', 'desc')
                        ->paginate(10); //paginate($perPage)
    $data = compact('posts');
    return view('posts.index', $data);
}

// resources/views/posts/index.blade.php
<div class="text-center">
    {!! $posts->render() !!}
</div>
```

隨機文章

- Laravel 的 Collection 可以直接幫我們從中隨機取出一至數個內容，只需要呼叫 `random()` 即可

```
// app/Http/Controllers/PostsController.php
public function random()
{
    $post = \App\Post::all()->random(); //random($number)

    $data = compact('post');

    return view('posts.show', $data);
}
```

Model 的時間戳記

- 在 Eloquent 的 Model 設計裡，每一個 Model 都會有兩個時間戳記的欄位：`created_at` 及 `updated_at`
- 當 Model 新建立時，`created_at` 和 `updated_at` 都會寫入當下的時間。但當 Model 更新時，則只會將當下的時間更新至 `updated_at`，這樣就可以做到紀錄 Model 建立及最後一次更新的時間戳記
- Eloquent 預設就會將這兩個時間欄位轉換成 Carbon 物件，也就是說，當我們在輸出這些欄位時，可以直接使用 Carbon 物件的 API 來做日期時間的轉換

常用 Carbon API

- Carbon 是 Modern PHP 非常好用的日期時間套件，可以非常方便的輸出、轉換、操作 PHP 的日期與時間，常用的 API 有：

```
- $model->created_at->toDateString();
- $model->created_at->toTimeString();
- $model->created_at->toDateTimeString();
- $model->created_at->format('Y-m-d H:i:s');
- $model->created_at->diffForHumans();
```

實作專案 CRUD

實作 CRUD

- 依據工作坊網站企劃書，實作不同 resource 的 CRUD 如下：
  - 針對文章內容做 CRUD
  - 針對留言做 CR
- 提示：
  - 使用 Form Builder 並綁定 Model 產生 DOM
  - 記得要設定 ServiceProvider、Aliases 及用 {!! 輸出

Eloquent 延伸用法

- 在 隨機文章 頁裡使用 `Model::random()` 取得隨機的一篇文章，再把該文章傳至 View 進行顯示
- 在有文章列表的頁面裡，加上 `Model::paginate()` 讓 Laravel 自動幫我們完成分頁查詢，並在 View 裡面產生分頁選單連結

存檔點

- 試著把現在已經可以運作的程式碼加入版本控制內
- 流程提醒：
  - working directory > staging area > commit

單元總結

- 在這個單元裡我們學到了些什麼？
  - CRUD 基本觀念及慣例
  - 使用 Form Builder 整合 Model 產生 DOM
  - Eloquent 的額外用法
  - 實作專案所需的 CRUD 功能

Q & A

歡迎提問討論