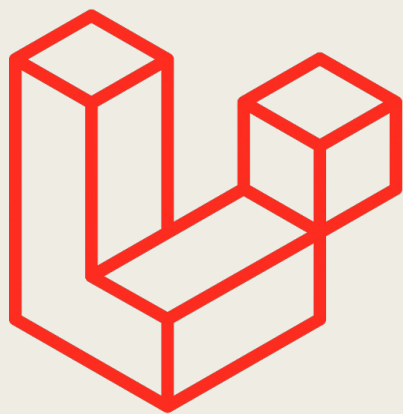


# 語系

朱克剛



# 產生語系目錄

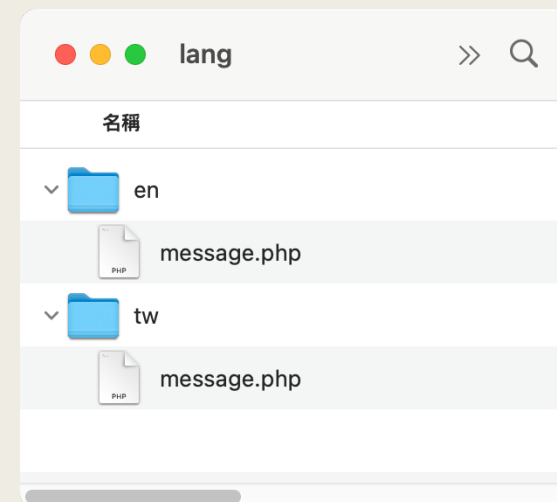
- 指令：
  - *php artisan lang:publish*
- 會在專案下建立 lang/en 目錄
- 有兩種語系設定方式
  - *php 形式的語系檔*
  - *json 格式的語系檔*

# PHP 語系檔

- 在 lang 目錄下建立 tw 目錄 ( 名稱可任意 )
- 在 en 與 tw 目錄下各放一個 message.php ( 檔名可任意 )

```
<?php
return [
    'home' => 'Home',
    'button' => 'Button',
    'label' => 'Label'
];
```

```
<?php
return [
    'home' => '首頁',
    'button' => '按鈕',
    'label' => '標籤'
];
```



# 使用 – 以 route 為例

- 使用 `__()` 函數來載入語系檔中的內容，格式為 `__(語系檔檔名.key)`

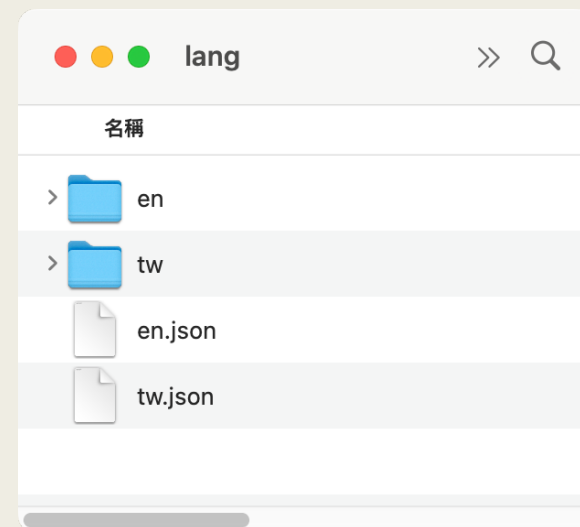
```
Route::get('/{locale}/php/{key}', function($locale, $key) {  
  
    App::setLocale($locale);  
    return __("message.{key}");  
  
})->whereIn('locale', ['en', 'tw']);
```

- 試試網址：

- <http://localhost/laravel/public/en/php/button>
- <http://localhost/laravel/public/tw/php/button>

# JSON 格式的語系檔

- 在 lang 目錄下建立 en.json 與 tw.json



```
{  
  "welcome": "Welcome",  
  "hello": "Hello"  
}
```

```
{  
  "welcome": "歡迎",  
  "hello": "你好"  
}
```

# 使用

- 一樣使用 `__()` 函數，但只要傳入 `key` 即可，例如 `__('welcome')`

```
Route::get('/{locale}/json/{key}', function($locale, $key) {  
    App::setLocale($locale);  
    return __('key'); // 注意這裡不需要加上檔名  
})->whereIn('locale', ['en', 'tw']);
```

# 使用 JSON 好處

- 比較容易找到自動翻譯的資源，例如 <https://translate.i18next.com>
  - 中文要輸入「zh-TW」
- 或是：<https://github.com/while1618/i18n-auto-translation>
- 缺點：只能有一個語系檔

# 預設語系設定

- 位於 config/app.php 中

```
'locale' => 'en',
```