

使用者認證

范聖佑 Shengyou Fan
新北市樹林國小 (2015/07/11)

```
1  /**
2   * Display the specified resource.
3   *
4   * @param int $id
5   * @return Response
6   */
7  public function show($id)
8  {
9      $post = Post::with('comments')->where('id', $id)->first();
10
11      return View::make('post.show')->with('post', $post);
12  }
```

單元主題

- 了解使用者認證機制的基本元素
- 介紹 Laravel 內建的使用者認證機制
- 如何在專案內整合 Laravel 內建的使用者認證機制
- 示範實作專案使用者認證功能

認證機制

使用者認證

- 網路應用程式的使用者認證機制，一般來說至少會
有以下幾個功能：
 - 使用者註冊
 - 使用者登入
 - 使用者登出
 - 忘記密碼 (寄出密碼重設信)
 - 重設密碼

Laravel 的認證機制

- Laravel 預設就提供了完整的使用者認證機制，上述列出的 5 大功能皆已內建，得來全不費功夫！
- 在 5.0 版預設就將使用者認證機制安裝好，若要移除的話，需要用 `artisan fresh` 指令將其移除
- 而在 5.1 版裡，則是只提供核心程式 (Middleware、Trait)、Model 及 Controller 的實作，至於 Route、View 等則需要手動撰寫
- 簡單來說，在 5.1 版只要將 Route 對應至 Laravel 提供的 Auth 系列 Controller，並將 View 補齊，就可以使用內建的認證機制

Laravel 認證相關的檔案

- 關於認證相關的設定檔放在 `config/auth.php` 內
- 預設好的 `User` Model 放在 `app/` 底下
- 預設好有兩個 migration 放在 `database/migrations` 內，會幫我們建立 `users` 及 `password_resets` 兩個資料表
- 預設有兩個 Controller 放在 `app/Http/Controllers/Auth` 底下：`AuthController` 及 `PasswordController`

使用者註冊

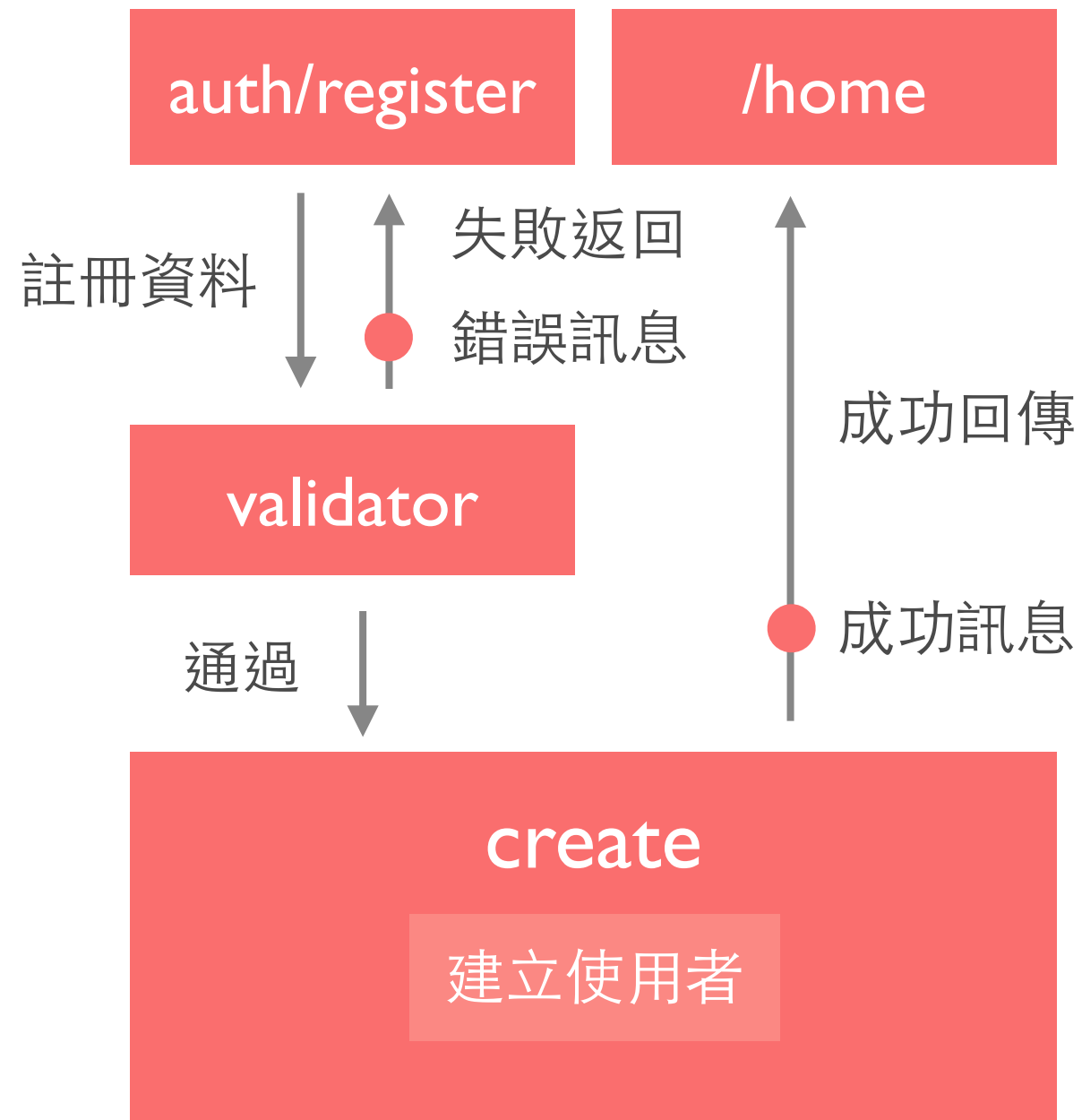
先建立 user 資料表

- Laravel 5.1 版有內建使用者認證的 migration 檔案，會自動幫我們建立好 `users` 資料表
- 若沒有其他欄位要增加的話，在實作前先把這個資料表透過 `php artisan migrate` 建立起來
- 若 `user` 需要有額外的欄位需要紀錄，則可以先修改 migration 檔的內容再進行 migrate；也可以新建一個 migration 把要擴充的欄位增加上去

user 資料表

- Laravel 的 `User` 是以 `email` 做為帳號唯一值，各欄位名稱及用途如下：
 - `id`：資料表 auto-increment 的 key 值
 - `name`：使用者名稱
 - `email`：使用者電子郵件，也同時做為帳號認證用
 - `password`：使用者密碼，經 `bcrypt` 加密後儲存，資料格式請設 `VARCHAR 60`
 - `remember_token`：認證時做為是否記住我的 token 值，資料格式至少 `VARCHAR 100`，在 migration 檔內可用 `$table->rememberToken()` 增加這個欄位

註冊流程示意圖



Laravel 的註冊流程

- 使用者先至 `/auth/register` 頁面填寫註冊表單，表單上有 **名字**、**Email**、**密碼**、**確認密碼** 四個欄位 (確認密碼欄位的名稱請設定為 `password_confirmation` 以利表單驗證時欄位對照用)
- 表單送出後，先經過 `AuthController@validator` 驗證表單資料，若失敗則返回、若成功則建立新的使用者
- 驗證表單成功後，`AuthController@create` 會依欄位建立新的使用者
- 所有動作完成後，返回至 `/home`

設定註冊 Routes

- Laravel 5.1 已經幫我們將 Controller 準備好，僅需要在 Route 裡指定路徑至對應的 Controller 即可
- 要注意 Controller 要加 **Auth** 的 **namespace**，並且注意對應到的 Controller 方法不可錯
- 若有需要，可自定 Route 名稱，方便其他頁面以 **route()** 連結

```
// app/Http/routes.php
Route::get('auth/register', ['as' => 'register.index',
    'uses' => 'Auth\AuthController@getRegister']);

Route::post('auth/register', ['as' => 'register.process',
    'uses' => 'Auth\AuthController@postRegister']);
```

實作註冊表單

- Laravel 5.1 預設沒有提供對應的 Views，不過官方文件內有示範如何撰寫表單原型，可以直接參考來改
- View 所預設放置的位置及檔名為：**resources/views/auth/register.blade.php** (請務必依照預設命名檔案並放置在對應的位置)

```
// resources/views/auth/register.blade.php
{!! Form::open(['route' => 'register.process',
                'method' => 'post']) !!}
    {!! Form::text('name', null, ['id' => 'name']) !!}
    {!! Form::email('email', null, ['id' => 'email']) !!}
    {!! Form::password('password', null, ['id' => 'password']) !!}
    {!! Form::password('password_confirmation', null,
                      ['id' => 'password_confirmation']) !!}
    {!! Form::submit('登入') !!}
{!! Form::close() !!}
```

修改預設設定

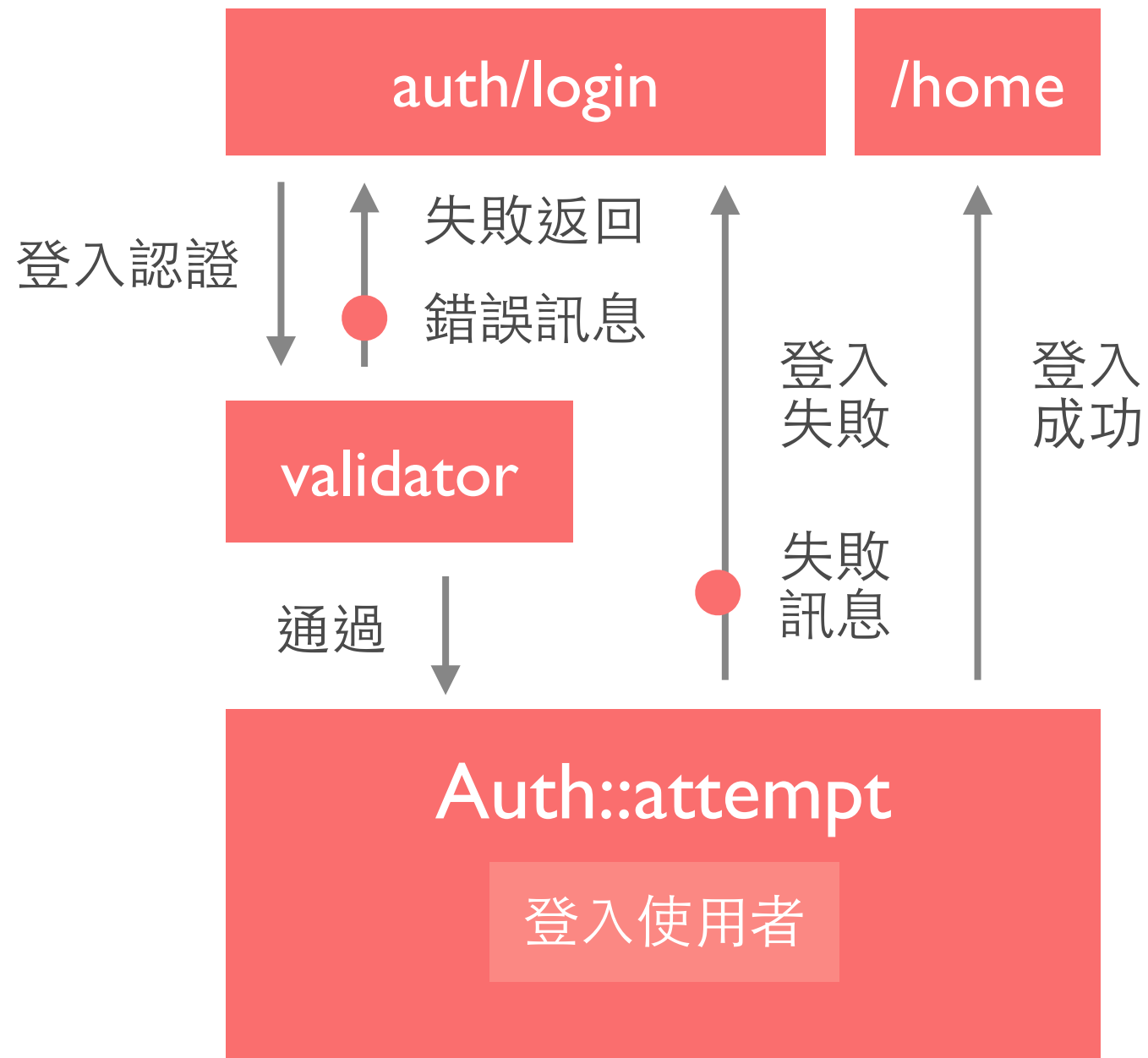
- 假如使用者資料表有需要額外的欄位，記得在 `AuthController` 裡的 `validator` 及 `create` 加上對應的驗證及寫入欄位的設定，而 `User Model` 裡的 `fillable` 也要將欄位打開
- 若註冊完成後，不想要導到 `/home`，可以在 `AuthController` 內設定一個 `$redirectPath` 屬性，將要重新導向的網址以字串設定即可

註冊流程核心程式

- 關於使用者註冊的功能，Laravel 預設提供了 `AuthController` 處理所有的動作，裡面的程式碼不多，因為大多數的動作都直接使用 `Illuminate\Foundation\Auth\AuthenticatesAndRegistersUsers` Trait 實作
- `AuthenticatesAndRegistersUsers` 裡又分成兩個 Trait，其中 `Illuminate\Foundation\Auth\RegistersUsers` 包含了所有關於註冊的動作，想了解 Laravel 如何實作註冊動作，可以追一下原始碼

使用者登入/登出

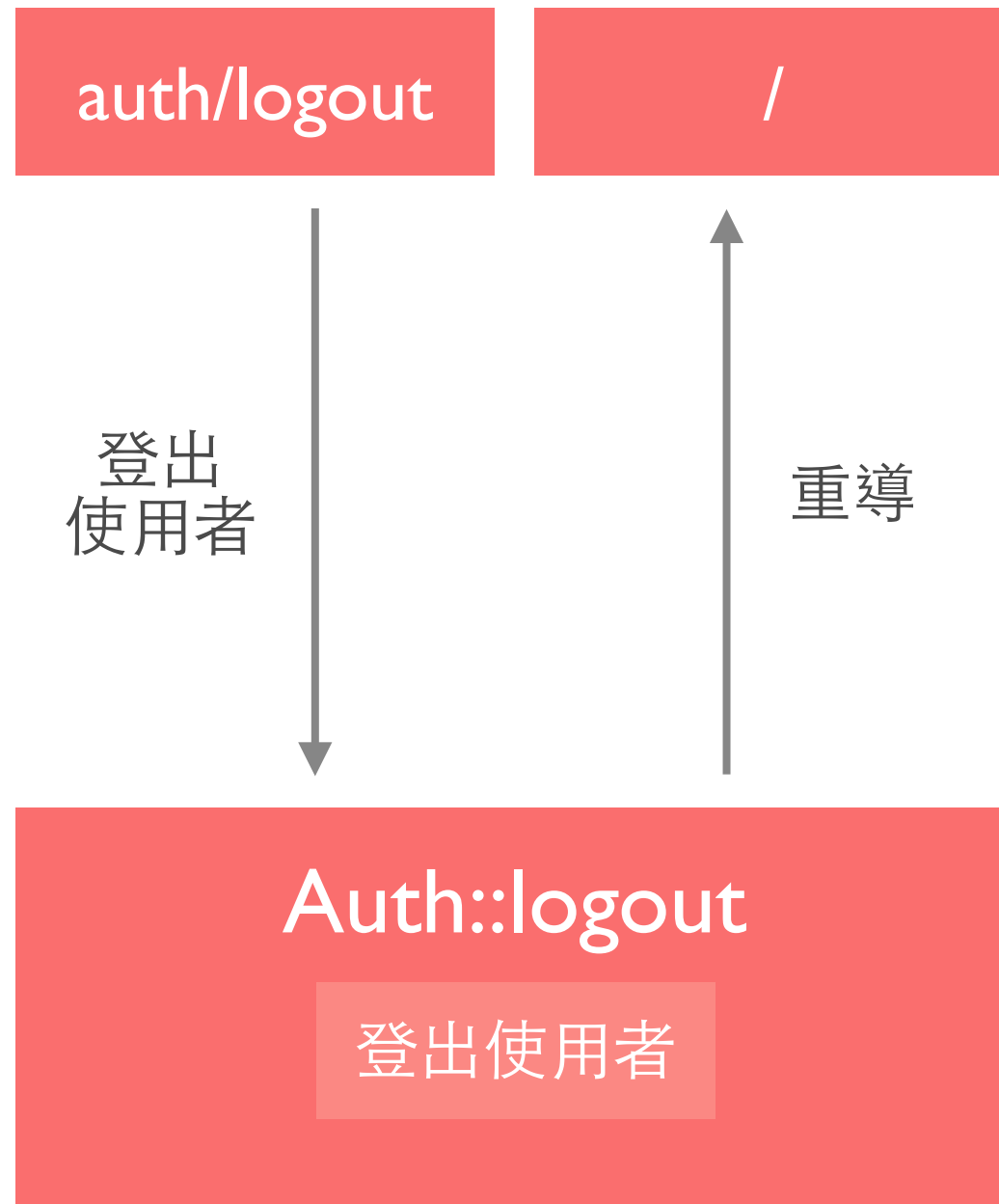
登入流程示意圖



Laravel 的登入流程

- 使用者先至 `/auth/login` 頁面填寫登入表單，表單上有 `Email`、`密碼` 兩個欄位及一個 `記住我` 的選勾欄位
- 表單送出後，`AuthController` 會自動用 `email` 及 `password` 兩個欄位的值與資料庫內的資料做比對
- 若失敗則返回登入頁、若成功則導至 `/home`

登出流程示意圖



Laravel 的登出流程

- 將使用者導至 `/auth/logout` 頁面
- `AuthController` 會將現在登入的使用者登出
- 所有動作完成後，返回至 `/`

設定登入/出 Routes

- Laravel 5.1 已經幫我們將 Controller 準備好，僅需要在 Route 裡指定路徑至對應的 Controller 即可
- 要注意 Controller 要加 **Auth** 的 **namespace**，並且注意對應到的 Controller 方法不可錯
- 若有需要，可自定 Route 名稱，方便其他頁面以 **route()** 連結

```
// app/Http/routes.php
Route::get('auth/login', ['as' => 'login.index', 'uses' =>
    'Auth\AuthController@getLogin']);
Route::post('auth/login', ['as' => 'login.process', 'uses'
=> 'Auth\AuthController@postLogin']);

Route::get('auth/logout', ['as' => 'logout.process', 'uses'
=> 'Auth\AuthController@getLogout']);
```

實作登入表單

- Laravel 5.1 預設沒有提供對應的 Views，不過官方文件內有示範如何撰寫表單原型，可以直接參考來改
- View 所預設放置的位置及檔名為：**resources/views/auth/login.blade.php** (請務必依照預設命名檔案並放置在對應的位置)

```
// resources/views/auth/login.blade.php
{!! Form::open(['route' => 'login.process', 'method' => 'post']) !!}

    {!! Form::email('email', null, ['id' => 'email']) !!}
    {!! Form::password('password', null, ['id' => 'password']) !!}
    {!! Form::checkbox('remember', 1, true, ['id' => 'remember']) !!}

    {!! Form::submit('登入') !!}

{!! Form::close() !!}
```

修改預設設定

- 若想要修改登入失敗的訊息 (不是表單驗證的錯誤訊息)，可以在 `AuthController` 裡覆寫 `getFailedLoginMessage` 方法裡的回傳字串
- 若登入完成後，不想要導到 `/home`，可以在 `AuthController` 內設定一個 `$redirectPath` 屬性，將要重新導向的網址以字串設定即可
- 若登出完成後，不想要導到 `/`，可以在 `AuthController` 內設定一個 `$redirectAfterLogout` 屬性，將要重新導向的網址以字串設定即可

登入/出流程核心程式

- AuthController 使用了 Illuminate\Foundation\Auth\AuthenticatesAndRegistersUsers Trait，而 AuthenticatesAndRegistersUsers 裡又分成兩個 Trait，其中 Illuminate\Foundation\Auth\AuthenticatesUsers 包含了所有關於登入/出的動作，想了解 Laravel 如何實作登入/出動作，可以追一下原始碼

忘記密碼

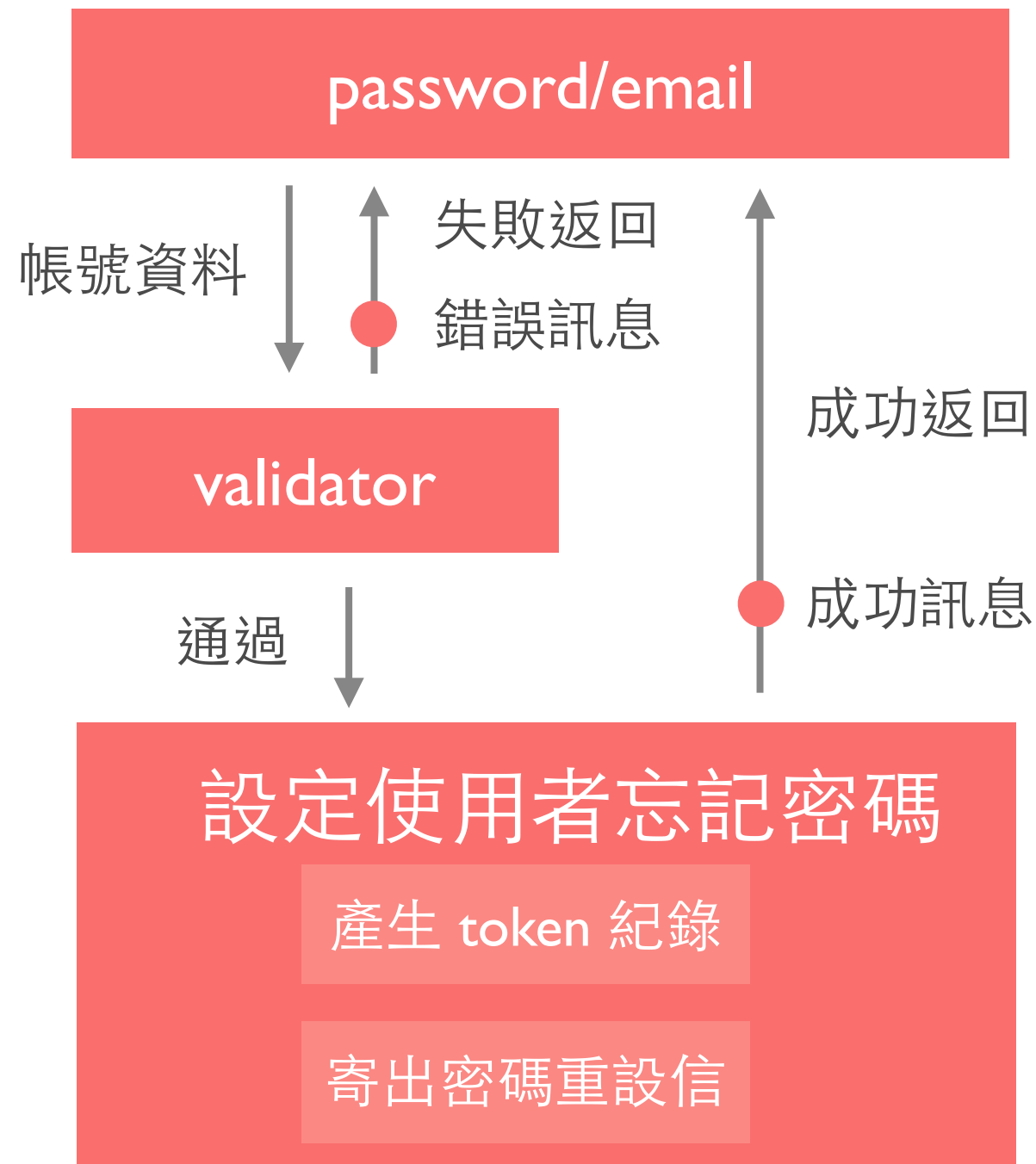
先建立資料表

- Laravel 5.1 版有內建密碼重設的 migration 檔案，會自動幫我們建立好 `password_resets` 資料表
- 在實作前要把這個資料表透過 `php artisan migrate` 建立起來

password_resets 資料表

- 當 Laravel 要讓使用者重設密碼時，會將重設密碼的 token 值存放在這個資料表，做為使用者可否重設密碼的認證：
 - email：使用者帳號
 - token：由 Laravel 產生的 token 值，做為比對用
 - created_at：紀錄申請重設的時間，預設為 1 小時，若超過時間則要重新申請。

忘記密碼流程示意圖



Laravel 的忘記密碼流程

- 使用者先至 `/password/email` 頁面忘記密碼表單，表單上只有 `Email` 一個欄位
- 表單送出後，先經過 `PasswordController@postEmail` 驗證表單資料，若失敗則返回
- 驗證表單成功後，`PasswordController` 會產生重設密碼 `token` 並紀錄在 `password_resets` 資料表內，並將重設密碼連結 `email` 給使用者
- 所有動作完成後，返回至 `/password/email`

設定忘記密碼 Routes

- Laravel 5.1 已經幫我們將 Controller 準備好，僅需要在 Route 裡指定路徑至對應的 Controller 即可
- 要注意 Controller 要加 **Auth** 的 **namespace**，並且注意對應到的 Controller 方法不可錯
- 若有需要，可自定 Route 名稱，方便其他頁面以 **route()** 連結

```
// app/Http/routes.php
```

```
Route::get('password/email' , ['as' => 'forgetpassword.index' ,  
'uses' => 'Auth\PasswordController@getEmail']);
```

```
Route::post('password/email', ['as' => 'forgetpassword.process',  
'uses' => 'Auth\PasswordController@postEmail']);
```

實作忘記密碼表單

- Laravel 5.1 預設沒有提供對應的 Views，不過官方文件內有示範如何撰寫表單原型，可以直接參考來改
- View 所預設放置的位置及檔名為：`resources/views/auth/password.blade.php` (請務必依照預設命名檔案並放置在對應的位置)
- 重設密碼完成後，會把成功訊息以 `status` 變數放在 `session` 裡，可在 view 上面印出來

```
// resources/views/auth/password.blade.php
{!! Form::open(['route' => 'forgetpassword.process',
                'method' => 'post']) !!}
    {!! Form::email('email', null, ['id' => 'email']) !!}
    {!! Form::submit('忘記密碼') !!}
{!! Form::close() !!}
```

設定 Email

- 確認完使用者身份後，Laravel 會幫我們寄出一封密碼重設信，所以要先設定 Email：
 - 設定 `config/mail.php` 裡的 `from` 區段內的寄件人資訊，包括 `address` 及 `name`
 - 設定 `.env` 裡的相關的 Email 設定值：
 - ✓ `MAIL_DRIVER` - 一般是 SMTP
 - ✓ `MAIL_HOST` - Email 使用的 SMTP 主機
 - ✓ `MAIL_PORT`、`MAIL_ENCRYPTION` - 主機使用的 port 及加密方式
 - ✓ `MAIL_USERNAME`、`MAIL_PASSWORD` - Email 帳號、密碼

設定重設密碼信內容

- Laravel 內建帶有 `resources/views/emails/password.blade.php` 會用這個 view 當做重設密碼信的內容
- 若想要修改信件的內容，直接修改這個 view 裡的 HTML 即可

```
// resources/views/emails/password.blade.php
```

```
點選此連結來重設您的密碼：{{ url('password/reset/'.$token) }}
```

修改預設設定

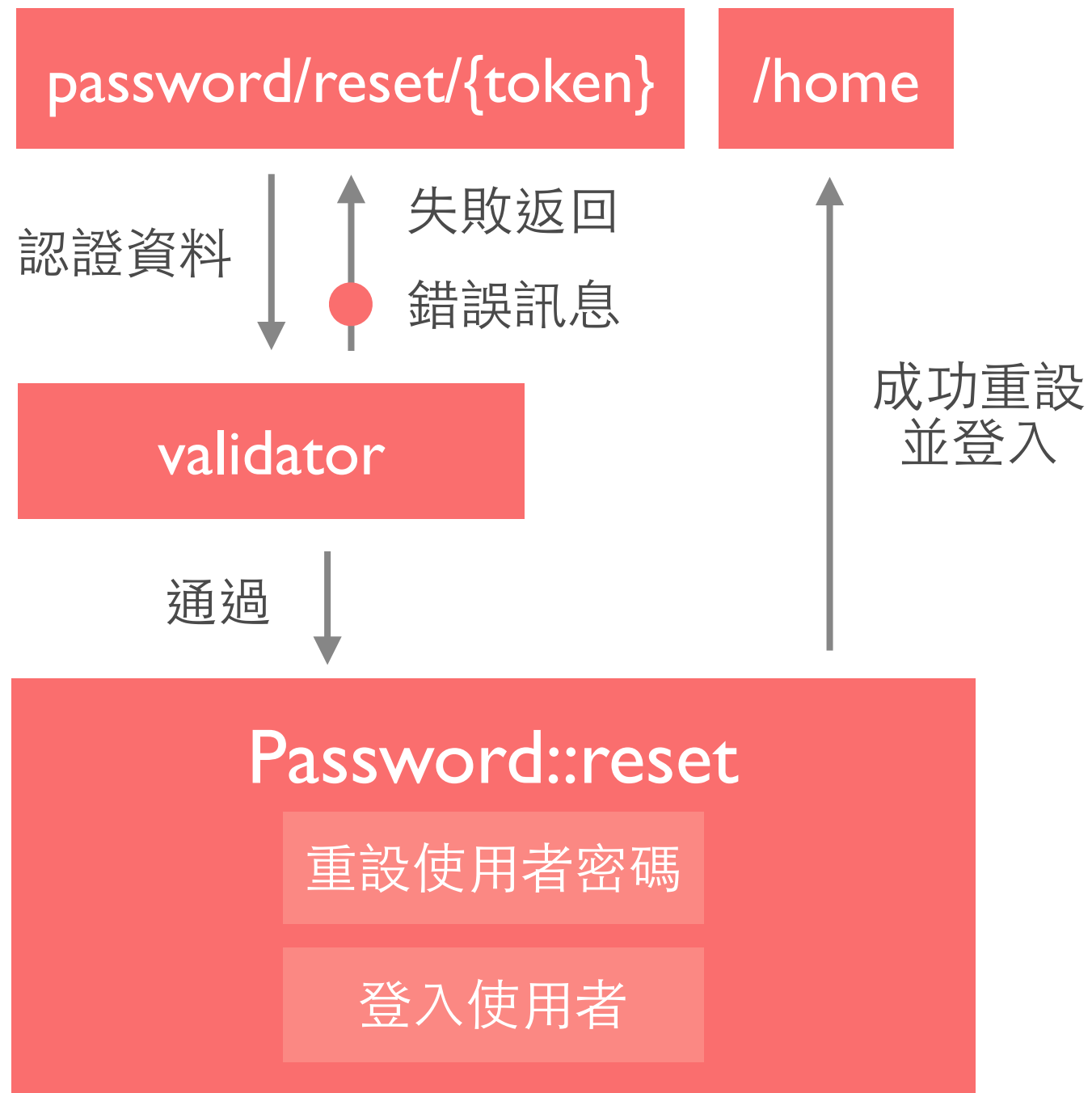
- 假如想要修改重設密碼信的信件主旨，可以在 `PasswordController` 裡設定一個 `$subject` 屬性，以字串設定信件主旨即可
- 所有跟密碼重設的相關提示文字，都以多國語系字串存在 `resources/lang/{code}/passwords.php` 內
- 其他更多重設密碼的設定，都紀錄在 `config/auth.php` 裡的 `password` 區段，包括重設密碼信使用的 `view` (`emails.password`)、重設密碼使用的資料表名稱 (`password_resets`) 以及重設密碼的有效時間 (`reminder.expire` 以分鐘計)

忘記密碼流程核心程式

- 關於忘記密碼的功能，Laravel 預設提供了 `PasswordController` 處理所有的動作，裡面大多數的動作都直接使用 `Illuminate\Foundation\Auth\ResetPasswords Trait` 實作，想了解 Laravel 如何實作忘記密碼動作，可以追一下原始碼

重設密碼

重設密碼流程示意圖



Laravel 的重設密碼流程

- 使用者從重設密碼信裡取得 `/password/reset/{token}` 頁面的網址，點擊網址後進入重設密碼頁面，若 token 無效則回傳 404
- 重設密碼表單上有 **Email**、**密碼**、**確認密碼** 三個欄位 (確認密碼欄位的名稱請設定為 `password_confirmation` 以利表單驗證時欄位對照用) 及一個 **token** 的隱藏欄位
- 表單送出後，先經過 `PasswordController@postReset` 驗證表單資料，若失敗則返回
- 驗證表單成功後，`PasswordController` 會將新密碼重新加密後儲存，並將使用者登入
- 所有動作完成後，返回至 `/home`

設定重設密碼 Routes

- Laravel 5.1 已經幫我們將 Controller 準備好，僅需要在 Route 裡指定路徑至對應的 Controller 即可
- 要注意 Controller 要加 **Auth** 的 **namespace**，並且注意對應到的 Controller 方法不可錯
- 若有需要，可自定 Route 名稱，方便其他頁面以 **route()** 連結

```
// app/Http/routes.php
```

```
Route::get('password/reset/{token}', ['as' => 'resetpassword.index',  
'uses' => 'Auth\PasswordController@getReset']);
```

```
Route::post('password/reset', ['as' => 'resetpassword.process',  
'uses' => 'Auth\PasswordController@postReset']);
```

實作重設密碼表單

- Laravel 5.1 預設沒有提供對應的 Views，不過官方文件內有示範如何撰寫表單原型，可以直接參考來改
- View 所預設放置的位置及檔名為：**resources/views/auth/reset.blade.php** (請務必依照預設命名檔案並放置在對應的位置)

```
// resources/views/auth/reset.blade.php
{!! Form::open(['route' => 'resetpassword.process', 'method' =>
'post']) !!}

    {!! Form::hidden('token', $token) !!}
    {!! Form::email('email', null, ['id' => 'email']) !!}
    {!! Form::password('password', null, ['id' => 'password']) !!}
    {!! Form::password('password_confirmation', null,
                        ['id' => 'password_confirmation']) !!}
    {!! Form::submit('重設密碼') !!}

{!! Form::close() !!}
```


修改預設設定

- 若重設密碼完成後，不想要導到 `/home`，可以在 `PasswordController` 內設定一個 `$redirectTo` 屬性，將要重新導向的網址以字串設定即可

重設密碼流程核心程式

- 關於重設密碼的功能，Laravel 預設提供了 `PasswordController` 處理所有的動作，裡面大多數的動作都直接使用 `Illuminate\Foundation\Auth\ResetsPasswords Trait` 實作，想了解 Laravel 如何實作重設密碼動作，可以追一下原始碼

認證相關實作

確認目前登入狀態

- Laravel 提供 **Auth Facade** 可以讓我們在任何地方確認使用者目前的登入狀態，回傳值是 **boolean**：
 - `\Auth::check()`

取得登入使用者資訊

- 當使用者成功登入系統後，Laravel 提供 2 種方式可以讓我們很方便的取得目前登入使用者的資訊：
 - `$user = \Auth::user();`
 - `$user = $request->user();`

限制僅供登入後瀏覽

- 如何保護頁面限制只有登入的使用者可以瀏覽？有兩種方式：

- 在 Route 上設定 middleware

```
// app/Http/routes.php
Route::group('users/profile', ['middleware' => 'auth', function()
{
    // 只有登入後的使用者才可以通過
}]);
```

- 在 Controller 的 construct 裡設定 middleware

```
// app/Http/Controllers/UsersController.php
public function __construct()
{
    $this->middleware('auth');
}
```

若已經登入自動跳轉

- 若在某些頁面要讓已經登入的使用者瀏覽時自動跳轉 (如登入頁、忘記密碼、重設密碼頁等)，可在 Controller 上設定 **guest** 的 middleware

```
// app/Http/Controllers/Auth/PasswordController.php
public function __construct()
{
    $this->middleware('guest');
}
```

- 若要修改跳轉頁的網址，請修改 app/Http/Middleware/RedirectIfAuthenticated.php 裡的路徑

```
// app/Http/Middleware/RedirectIfAuthenticated.php
if ($this->auth->check()) {
    return redirect('/');
}
```

單元總結

- 在這個單元裡我們學到了些什麼？
 - 標準網路應用程式的使用者認證實作
 - Laravel 5.1 提供的使用者認證功能
 - 示範專案使用者認證實作方式



歡迎提問討論