Middleware & 身分驗證

勤益資管 廖文忠

MIDDLEWARE

Middleware

- 作用
  - before middleware vs. after middleware
  - 用以對進入AP的request (before entering AP)或離開AP的response (after leaving AP)進行檢查或過濾的操作
  - 用以對路由進行保護
- Laravel內建的middleware主要：
  - 身分驗證 (Authentication)
  - CSRF檢查與保護
  - logging middleware: log (紀錄) all incoming requests to your application
  - …
- 參考資料：
  - https://laravel.com/docs/10.x/middleware

自訂Middleware

- *artisan make:middleware* [中介層程式名稱]
- app/Http/Middleware：存放middleware的資料夾

- https://laravel.com/docs/10.x/middleware#defining-middleware

自訂Middleware範例

- 命令：php artisan make:middleware EnsureTokenIsValid
- EnsureTokenIsValid 會是一個類別(位置: app/Http/Middleware)：

```
public function handle($request, Closure $next)
    {
        if ($request->input('token') !== 'my-secret-token') {
            return redirect('/');
        }
        return $next($request);
    }
```

-
```
Route::get('/test', function () {
    return 'pass';
}) ->middleware(EnsureTokenIsValid::class);
```

- 測試URL : /test
- 測試URL : /test?token=my-secret-token

Middleware註冊

- app/HTTP/kernel.php
  - **$middleware**: These middleware are run during every request to your application. (個別作用為何?)
  - **$middlewareGroup**: Middleware Groups (多個Middleware的集合)，預設有web & api 兩組：
    - web (for web.php的路由), api (for api.php的路由)
  - **$middlewareAliases**: 定義middleware的別名，方便將middleware指定給路由或控制器，例如：
    - auth, auth.basic, guest, verified, …

- https://laravel.com/docs/10.x/middleware#registering-middleware

內建middleware alias (1/3)

- ‘auth’:檢查是否已登入(身分是否已驗證)，若是，繼續，否則轉向/login
  - 'auth' => \App\Http\Middleware\Authenticate
  - 使用範例：以下路由或控制器會先經過此middleware：
    - /dashboard, /profile等路由, /logout, …
    - /chirps等路由 (Chirper)
    - TaskController (quickstart2)

內建middleware alias (2/3)

- 'guest' ：檢查是否為訪客，若非，轉向 RouteServiceProvider::HOME
  - 'guest' => \App\Http\Middleware\RedirectIfAuthenticated
  - 使用範例：以下路由會先經過此middleware ：
    - /login, /register, … (請參考routes/auth.php)
  - 此middleware可傳入身分驗證的guard，例如:
    - guest:admin
    - guest:web

內建middleware alias (3/3)

- 'verified' 檢查使用者是否已經使用email驗證，若無，會寄出email要求使用者收email驗證
  - 'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified
  - Chirper範例：/dashboard, /chirps等路由用之，但需：
    - User Model需 implements MustVerifyEmail，且
    - 完成mail相關設定(可正常寄送信件)
  - 參考資料: https://blog.devgenius.io/laravel-enable-user-registration-email-verification-and-customize-email-templates-b994299ab27d
- 'throttle' 檢查訪問頻率是否超過上限 (可避免DoS攻擊?)

內建Middleware Groups

- 'web': 用於所有web路由 (routes/web.php)
  - VerifyCsrfToken: 檢查CSRF token是否正確
  - EncryptCookies (?)
  - StartSession(?)
  - …
  - SubstituteBindings(?)
- 'api' :用於所有api路由 (routes/api.php)
  - 'throttle:api' (?)
  - SubstituteBindings (?)

AUTHENTICATION (身分驗證)

config\auth.php
(Auth基本設定：**Guard**)

```
'defaults' => [
        'guard' => 'web',      //預設web guard
        'passwords' => 'users',
    ],

 'guards' => [        //web guard和 api guard
        'web' => [
            'driver' => 'session',
            'provider' => 'users',
        ],
        'api' => [
            'driver' => 'token',
            'provider' => 'users',
            'hash' => false,
        ],
    ],
```

```
'providers' => [
        'users' => [
            'driver' => 'eloquent',
            'model' => App\Models\User::class,
        ],

        // 'users' => [
        //     'driver' => 'database',
        //     'table' => 'users',
        // ],
    ],
```

Guard: 用來驗證使用者身分的“警衛”

身分驗證套件I：Laravel Breeze

- routes\auth.php: 定義所有身分驗證相關路由: login, register, …
- routes\web.php: 定義/profile相關路由, /dashboard
- app\Http\Controllers\Auth\: 所有身分驗證相關控制器，
  - AuthenticatedSessionController, RegisteredUserController, …
- app\Http\Controllers\ProfileController.php: 使用者維護基本資料的控制器
- app\Models\User.php: 身分驗證使用的User model
- resources\views\auth\: 所有身分驗證相關視圖
  - login.blade.php, register.blade.php, forget-password.blade.php, …
- resources\views\components\: 所有身分驗證相關視圖使用的元件
- resources\views\layouts\: 所有身分驗證相關視圖的主樣板與共同區塊，或稱為loyout元件。
  - app.blade.php, guest.blade.php, navigation.blade.php
- resources\views\profile\: 使用者基本資料維護時使用的視圖。

routes\auth.php

- GET|HEAD login..... login ›
  Auth**AuthenticatedSessionController@create**
- POST login.....
  Auth**AuthenticatedSessionController@store**
- POST logout..... logout ›
  Auth**AuthenticatedSessionController@destroy**

- Auth**AuthenticatedSessionController**
  - create(): 產生登入的表單(view): auth/login.blade.php
  - store(): 檢驗帳密、產生新的會談(session)、轉向到 RouteServiceProvider::HOME

routes\auth.php

- GET|HEAD register ..... register ›
  Auth\RegisteredUserController@create
- POST register .....
  Auth\RegisteredUserController@store

- Auth\RegisteredUserController
  - create(): 產生註冊的表單(view): auth/register.blade.php
  - store(): 檢驗註冊資料、新增使用者、發送使用者新增事件、登入新使用者、轉向到RouteServiceProvider::HOME

routes\auth.php

- GET|HEAD forgot-password ..... password.request › Auth\PasswordResetLinkController@create
- POST forgot-password ..... password.email › Auth\PasswordResetLinkController@store

routes\auth.php

- POST reset-password ..... password.store ›
  Auth\NewPasswordController@store
- GET|HEAD reset-password/{token} ..... password.reset ›
  Auth\NewPasswordController@create

routes\auth.php

- GET|HEAD verify-email ..... verification.notice ›
  Auth\EmailVerificationPromptController@__invoke
- GET|HEAD verify-email/{id}/{hash} ..... verification.verify ›
  Auth\VerifyEmailController@__invoke

- GET|HEAD confirm-password ..... password.confirm ›
  Auth\ConfirmablePasswordController@show
- POST confirm-password .....
  Auth\ConfirmablePasswordController@store

- POST email/verification-notification ..... verification.send ›
  Auth\EmailVerificationNotificationController@store
- PUT password ..... password.update ›
  Auth\PasswordController@update

routes\web.php

- GET|HEAD profile..... profile.edit ›
  ProfileController@edit
- PATCH profile..... profile.update ›
  ProfileController@update
- DELETE profile..... profile.destroy ›
  ProfileController@destroy

身分驗證套件II： Laravel\JetStream

- GET|HEAD login ..... login › Laravel\Fortify ›
  AuthenticatedSessionController@create
- POST login .... Laravel\Fortify ›
  AuthenticatedSessionController@store
- POST logout ..... logout › Laravel\Fortify ›
  AuthenticatedSessionController@destroy

Laravel\JetStream

- GET|HEAD register .... register › Laravel\Fortify › RegisteredUserController@create
- POST register .... Laravel\Fortify › RegisteredUserController@store

Laravel\JetStream

- POST reset-password ... password.update ›
  Laravel\Fortify › NewPasswordController@store
- GET|HEAD reset-password/{token} .... password.reset ›
  Laravel\Fortify › NewPasswordController@create

Laravel\JetStream

- GET|HEAD user/confirm-password ..... Laravel\Fortify › ConfirmablePasswordController@show
- POST user/confirm-password ..... password.confirm › Laravel\Fortify › ConfirmablePasswordController@store
- GET|HEAD user/confirmed-password-status ..... password.confirmation › Laravel\Fortify › ConfirmedPasswordStatusController@show
- POST user/confirmed-two-factor-authentication ..... two-factor.confirm › Laravel\Fortify › ConfirmedTwoFactorAuthenticationController@store
- PUT user/password ..... user-password.update › Laravel\Fortify › PasswordController@update
- GET|HEAD user/profile ..... profile.show › Laravel\Jetstream › UserProfileController@show
- PUT user/profile-information ..... user-profile-information.update › Laravel\Fortify › ProfileInformationController@update

Laravel\JetStream

- GET|HEAD two-factor-challenge ..... two-factor.login › Laravel\Fortify › TwoFactorAuthenticatedSessionController@create
- POST two-factor-challenge ..... Laravel\Fortify › TwoFactorAuthenticatedSessionController@store
- POST user/two-factor-authentication ..... two-factor.enable › Laravel\Fortify › TwoFactorAuthenticationController@store
- DELETE user/two-factor-authentication ..... two-factor.disable › Laravel\Fortify › TwoFactorAuthenticationController@destroy
- GET|HEAD user/two-factor-qr-code ..... two-factor.qr-code › Laravel\Fortify › TwoFactorQrCodeController@show
- GET|HEAD user/two-factor-recovery-codes ..... two-factor.recovery-codes › Laravel\Fortify › RecoveryCodeController@index
- POST user/two-factor-recovery-codes ..... Laravel\Fortify › RecoveryCodeController@store
- GET|HEAD user/two-factor-secret-key ..... two-factor.secret-key › Laravel\Fortify › TwoFactorSecretKeyController@show

Auth Facade與身分驗證

- **Auth**::attempt($credentials) ;
  - 以$credentials當中的email & password，驗證是否為合法使用者
  - 假設$credentials目前存放某使用者登入時輸入的email & password：

```
$credentials = [ 'email'       => $request->email,
                 'password' => $request->password  ];
```

- **Auth** ::guest()：檢查是否為訪客
- **Auth** ::check()：檢查是否已登入
- **Auth** ::login($user)：將$user設為已登入者
  - $user通常是一個User Model
- **Auth** ::logout() ：將連線(session)的登入者登出

Auth Facade與登入者相關資料擷取

- Auth::user() ：登入者的User Model
- Auth::user()->id：登入者的id
- Auth::user()-> name：登入者的name
- Auth::user()-> email：登入者的email
- $tasks = Auth::user()-> **tasks()**->get() ;
  - 在quickstart2當中，用以擷取登入者所有任務(Task)。
  - 在quickstart2當中，**tasks()**代表User hasMany Task。
- Auth::user()-> tasks()->create([…]);
  - quickstart2: 登入者新增自己的任務資料。

auth()輔助方法與Auth Facade

- auth()輔助方法可以完全取代Auth Facade，例如；
  - **auth()->**guest() 等同**Auth::**guest()
  - **auth()->**check() 等同**Auth::**check()
  - **auth()->**user() 等同**Auth::** user()

auth()輔助方法與Auth Facade

- 假若$request是一個Illuminate\Http\Request，物件則：
  - $request代表使用者所送出的HTTP request相關資訊
  - $request->user()代表送出HTTP request的使用者。
  - 若使用者已登入，則：
    - **$request->**user()也等同**Auth::**user()
  - **$request**->user()->**chirps()**->create($validated);
    - 可新增使用者的聊天訊息
  - $request相當好用，請參考Laravel文件與相關範例。

Multi-Guard Authentication

- 使用一個以上的Guards來驗證使用者身分。
- 不同身分的使用者或不同方式進入系統的使用者，以不同的Guard驗證其身分。
- 例如，如果你的系統有一般身分的Web使用者與系統管理員，且使用不同的資料表& Model，則可以用：
  - web guard: 驗證一般Web使用者身分，使用User model & users table
  - admin guard: 驗證系統管理員身分，使用Admin model & admins table
  - api guard: 使用系統提供的api進入系統的Web使用者，預設使用User model & users table

Multi-Guard Authentication

- 範例: multi-guard-auth
  - 一般使用者登入登出路由：
    - /login, /logout
  - **Admin使用者**登入登出路由：
    - /admin/login, /admin/logout
- 參考資料：
  https://www.laravelia.com/post/laravel-9-multi-auth-without-user-model-using-guard