

Rutas avanzadas

Laravel permite crear grupos de rutas para especificar opciones comunes a todas ellas, como por ejemplo un *middleware*, un prefijo, un subdominio o un espacio de nombres que se tiene que aplicar sobre todas ellas.

A continuación vamos a ver algunas de estas opciones, en todos los casos usaremos el método `Route::group`, el cual recibirá como primer parámetro las opciones a aplicar sobre todo el grupo y como segundo parámetro una clausula con la definición de las rutas.

Middleware sobre un grupo de rutas

Esta opción es muy útil para aplicar un filtro sobre todo un conjunto de rutas, de esta forma solo tendremos que especificar el filtro una vez y además nos permitirá dividir las rutas en secciones (distinguiendo mejor a que secciones se les está aplicando un filtro):

```
Route::group(['middleware' => 'auth'], function () {
    Route::get('/', function () {
        // Ruta filtrada por el middleware
    });

    Route::get('user/profile', function () {
        // Ruta filtrada por el middleware
    });
});
```

Grupos de rutas con prefijo

También podemos utilizar la opción de agrupar rutas para indicar un prefijo que se añadirá a todas las URL del grupo. Por ejemplo, si queremos definir una sección de rutas que empiecen por el prefijo `dashboard` tendríamos que hacer lo siguiente:

```
Route::group(['prefix' => 'dashboard'], function () {
    Route::get('catalog', function () { /* ... */ });
    Route::get('users', function () { /* ... */ });
});
```

También podemos crear grupos de rutas dentro de otros grupos. Por ejemplo para definir un grupo de rutas a utilizar en una API y crear diferentes rutas según la versión de la API podríamos hacer:

```
Route::group(['prefix' => 'api'], function()
{
    Route::group(['prefix' => 'v1'], function()
    {
        // Rutas con el prefijo api/v1
        Route::get('recurso', 'ControllerAPIv1@getRecurso');
        Route::post('recurso', 'ControllerAPIv1@postRecurso');
        Route::get('recurso/{id}', 'ControllerAPIv1@putRecurso');
    });

    Route::group(['prefix' => 'v2'], function()
    {
        // Rutas con el prefijo api/v2
        Route::get('recurso', 'ControllerAPIv2@getRecurso');
        Route::post('recurso', 'ControllerAPIv2@postRecurso');
        Route::get('recurso/{id}', 'ControllerAPIv2@putRecurso');
    });
});
```

De esta forma podemos crear secciones dentro de nuestro fichero de rutas para agrupar, por ejemplo, todas las rutas públicas, todas las de la sección privada de administración, sección privada de usuario, las rutas de las diferentes versiones de la API de nuestro sitio, etc.

Esta opción también la podemos aprovechar para especificar parámetros comunes que se recogerán para todas las rutas y se pasarán a todos los controladores o funciones asociadas, por ejemplo:

```
Route::group(['prefix' => 'accounts/{account_id}'], function () {
    Route::get('detail', function ($account_id) { /* ... */ });
    Route::get('settings', function ($account_id) { /* ... */ });
});
```