LARAVEL 5.8

enrutamiento

02/09/2019

INTRODUCCIÓN

Las rutas básicas de Laravel aceptan un **URI** y un **Closure**, proporcionando un método muy simple y expresivo para definir rutas:

```
Route::get('foo', function () {
    return 'Hello World';
});
```

ARCHIVOS DE RUTA PREDETERMINADOS:

Todas las rutas de Laravel se definen en sus archivos de ruta, que se encuentran en el directorio "**routes**". Estos archivos son cargados automáticamente por el framework. El archivo "**routes/web.php**" define rutas que son para su interfaz web. A estas rutas se les asigna el grupo de **middleware web**, que proporciona características como el estado de la sesión y la protección CSRF. Las rutas en "**routes/api.php**" no tienen estado y se les asigna el grupo de **middleware api**.

Para la mayoría de las aplicaciones, comenzará definiendo rutas en su archivo "**routes/web.php**". Se puede acceder a las rutas definidas en "**routes/web.php**" ingresando la URL de la ruta definida en su navegador. Por ejemplo, puede acceder a la siguiente ruta navegando a http: //your-app.test/user en su navegador:

```
Route::get('/user', 'UserController@index');
```

MÉTODOS DE ENRUTAMIENTO:

El enrutador le permite registrar rutas que responden a cualquier verbo **HTTP**:

```
Route::get($uri, $callback);

Route::post($uri, $callback);

Route::put($uri, $callback);

Route::patch($uri, $callback);

Route::delete($uri, $callback);

Route::options($uri, $callback);
```

A veces puede que necesite registrar una ruta que responda a múltiples verbos HTTP. Puede hacerlo utilizando el método **match**. O incluso puede registrar una ruta que responda a todos los verbos HTTP utilizando el método **any**:

```
Route::match(['get', 'post'], '/', function () {
    //
});

Route::any('/', function () {
    //
});
```

LISTAR RUTAS REGISTRADAS:

```
> php artisan route:list
```

LISTAR RUTAS REGISTRADAS DE FORMA COMPACTA:

```
> php artisan route:list --compact

> php artisan route:list -c
```

LIMPIAR RUTAS DEL CACHE:

```
> php artisan route:clear
```

REGISTRO DE RUTAS EN CACHE PARA MAYOR VELOCIDAD:

```
> php artisan route:cache
```

PROTECCIÓN CSRF:

Todos los formularios HTML que apuntan a rutas **POST**, **PUT** o **DELETE** que están definidas en el archivo de rutas **web** deben incluir un campo de token CSRF. De lo contrario, la solicitud será rechazada:

```
<form method="POST" action="/profile">
    @csrf
</form>
```

REDIRECCIONAR RUTAS:

Si está definiendo una ruta que redirige a otro **URI**, puede usar el método **Route::redirect**. Este método proporciona un atajo conveniente para que no tenga que definir una ruta completa o un controlador para realizar una redirección simple:

```
Route::redirect('/here', '/there');
```

ENRUTAR A UNA VISTA:

Si su ruta solo necesita devolver una vista, puede usar el método **Route::view**. Al igual que el método de redireccionamiento, este método proporciona un acceso directo simple para que no tenga que definir una ruta o controlador completo. El método de vista acepta un **URI** como su primer argumento y un nombre de vista como su segundo argumento. Además, puede proporcionar una matriz de datos para pasar a la vista como un tercer argumento opcional:

```
Route::view('/welcome', 'welcome');

Route::view('/welcome', 'welcome', ['name' => 'Taylor']);
```

PARAMETROS EN UNA RUTA:

Algunas veces necesitará capturar segmentos del URI dentro de su ruta. Por ejemplo, es posible que deba capturar la identificación de un usuario de la URL. Puede hacerlo definiendo parámetros de ruta:

```
Route::get('user/{id}', function ($id) {
    return 'User '.$id;
});
```

Puede definir tantos parámetros de ruta como requiera su ruta:

```
Route::get('posts/{post}/comments/{comment}', function ($postId, $commentId) {
    //
});
```

NOMBRES PARA LAS RUTAS:

Las rutas con nombre permiten la generación conveniente de URL o redireccionamientos para rutas específicas. Puede especificar un nombre para una ruta encadenando el método de nombre en la definición de ruta:

```
Route::get('user/profile', function () {
    //
})->name('profile');
```

También puede especificar nombres de ruta para acciones de controlador:

```
Route::get('user/profile', 'UserProfileController@show')->name('profile');
```