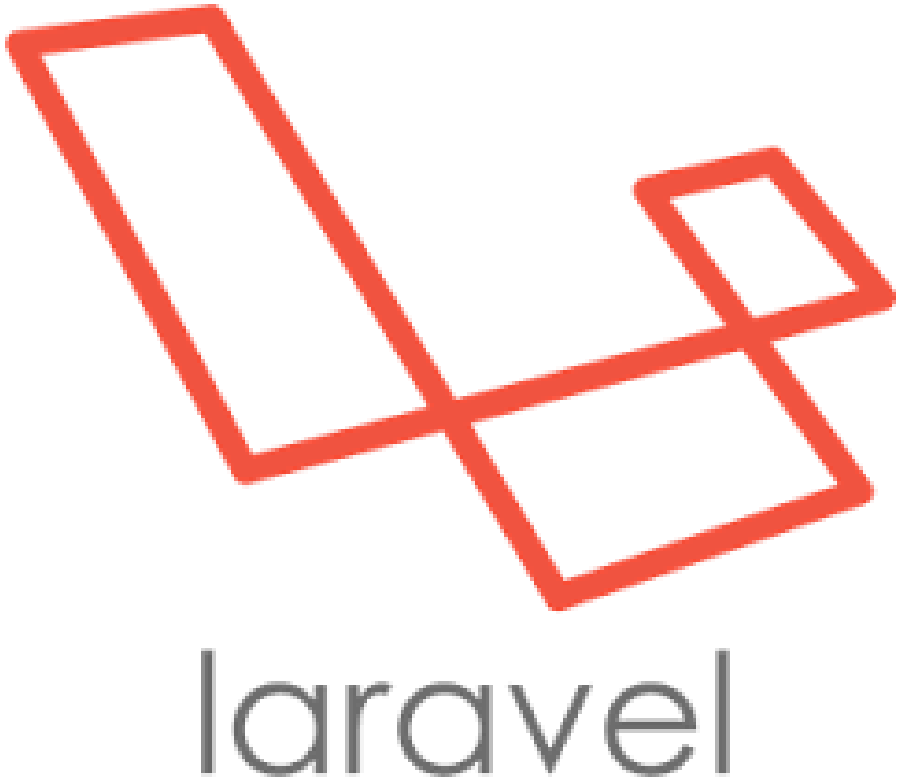
Curso de Laravel



Capítulo 4: Controladores

Laravel: Controladores

Controlador Básico

- Se ubican en el directorio app/Http/Controllers
- Se les añade el sufijo Controller ejemplo userController.php
- Ej: de controlador app/Http/Controllers/userController.php

```php
<?php
Namespace App\Http\Controllers;
use App\User;
use App\Http\Controllers\Controller;
Class UserController extends Controller
{   /**
     *  Mostrar información de un usuario.
     *  @param int $id
     *  @return Response
     */
    Public function showProfile($id)
    {
        $user = User::findOrFail($id);
        Return view('user.profile', ['user'=>$user]);
    }
}
```

Laravel: Controladores

Controladores

- Crear un nuevo controlador

```
php artisan make:controller clientesController
```

- Llamar a un controlador de una ruta

```
Route::get('user/{i}','UserController@showProfile');
```

- Generar una URL a una acción

```
$url = action('FooController@method');
```

- Generar un enlace a una accion desde una plantilla Blade

```
<a href="{{ action('FooController@method') }}">
¡Press here!</a>
```

Laravel: Controladores

Controladores implícitos

- Ruta

```
Route::controller('users','UserController');
```

- Controlador

```
class UserController extends BaseController
{
    public function getIndex(){
    //
    }
    public function postProfile(){
    //
    }
    public function anyLogin(){
    //
    }
}
```

Laravel: Controladores

Controladores implícitos

- Otros métodos

```
GetAdminProfile();
```

- Es equivalente a

```
users/admin-profile
```

- Missing Method

```
public function missingMethod($parameters = array())
{
    //
}
```

Ejercicio

Realizar lo siguiente

- Generar un nuevo controlador
- Enlazar controlador a la vista
- Enlazar el archivo de ruta al controlador

Laravel: Caché

Caché de rutas

- Crear caché

```
php artisan route:cache
```

- Borrar caché

```
php artisan cache:clear
```

*Se recomienda solo usar en producción

Laravel: Filtros

Filtros HTTP

- Filtran las peticiones HTTP
- Se ubican en el directorio `app/Http/Middleware`
- Puede ejecutar código antes o después de la procesar el código del controlador
- Definir un nuevo filtro

```
php artisan make:middleware MyMiddleware
```

Laravel: Filtros

Código Autogenerado

```php
<?php

namespace App\Http\Middleware;

use Closure;

class MyMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        return $next($request);
    }
}
```

Laravel: Filtros

Elementos que componen un filtro

- Función handle se encarga de implementar el filtro
- $request Recibe los parámetros de la petición
- $next Función que procesa posteriormente la petición

Laravel: Filtros

Ejemplo de implementación de un filtro

```php
public function handle($request, Closure $next)
{
    if($request->input("edad")<18){
        return redirect('home');
    }
    return $next($request);
}
```

Laravel: Filtros

Acciones posibles en un filtro

- Dejar que la petición continúe haciendo una llamada a `return $next($request);`
- Redirigir a otra ruta para impedir el acceso a la acción con `return redirect('home');`
- Lanzar una excepción o hacer una llamada al método abort para mostrar una página de error:
- `return abort(403,'No tienes acceso');`

Laravel: Filtros

Ejecución de un filtro antes de la acción

```php
public function handle($request, Closure $next)
{
   //código a ejecutar antes de la acción
   return $next($request);
}
```

Laravel: Filtros

Ejecución de un filtro después de la acción

```php
public function handle($request, Closure $next)
{
    $response = $next($request);
    //código a ejecutar después de la acción

    return $response;
}
```

Laravel: Filtros

Alcance de los filtros

Laravel permite definir el alcance de aplicación de los filtros

- Global (El filtro se ejecuta para todas las peticiones del proyecto)
- Asociado a rutas o grupos de rutas
- Asociado a un controlador
- Asociado a un método de un controlador

- Para asociar un filtro global, deberá registrarse el Middleware en el archivo app/Http/Kernel.php

Laravel: Filtros

Middleware Global

Para que el filtro se ejecute en todas las peticiones HTTP, se debe registrar en el array $middleware definido en la clase app/Http/Kernel.php

```
protected $middleware = [
\Illuminate\Foundation\Http\Middleware\CheckForMaintenanceMode::class,
\Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
\App\Http\Middleware\TrimStrings::class,
\Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
\App\Http\Middleware\MyMiddleware::class,
];
```

Laravel: Filtros

Middleware Asociado a rutas

Para asociar un filtro a las rutas se debe registrar en el array $routeMiddleware del archivo app/Http/Kernel.php

```
protected $routeMiddleware = [
 'auth' => \Illuminate\Auth\Middleware\Authenticate::class,
 'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
 'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,
 'can' => \Illuminate\Auth\Middleware\Authorize::class,
 'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
 'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
 'nuestro_filtro' => \App\Http\Middleware\MyMiddleware::class,
];
```

Laravel: Filtros

Middleware Asociado a rutas

Registrado en el archivo Kernel.php, se puede usar en el archivo de rutas

```
Route::get('/','Home')->middleware(['auth','otro_f']);

Route::get('profile',[
    'middleware'=>'auth',
    'uses'=>'UserController@showProfile'
]);

Route::get('dashboard',['middleware'=>'has_access',function(){
    //...
}]);
```

Laravel: Filtros

Middleware en un Controlador

Registrado en el archivo Kernel.php, se puede usar en el controlador

```
class UserController extends Controller
{
    /**
     *  Instantiate  a   new UserController   instance.
     *
     *  @return  void
     */
    public   function __construct()
    {
        //  Filtrar  todos   los  métodos
        $this->middleware('auth');
        //  Filtrar  soloestos   métodos...
        $this->middleware('log', ['only'  =>  ['fooAction',    'barAction']]);
        //  Filtrar  todos   los  métodos  excepto...
        $this->middleware('subscribed',   ['except'    =>  ['fooAction',    'barAction']]);
    }
}
```

Laravel: Filtros

Middleware Asociado a rutas

Registrado en el archivo Kernel.php, se puede usar en el archivo de rutas

```
Route::get('/','Home')->middleware(['auth','otro_f']);

Route::get('profile',[
    'middleware'=>'auth',
    'uses'=>'UserController@showProfile'
]);

Route::get('dashboard',['middleware'=>'has_access',function(){
    //...
}]);
```

Laravel: Filtros

Revisar los filtros asignados

- Ver los filtros en el resultado del siguiente comando

```
php artisan route:list
```

```
[michael@localhost inventario]$ php artisan route:list
+--------+----------+-----------+------+---------+--------------+
| Domain | Method   | URI       | Name | Action  | Middleware   |
+--------+----------+-----------+------+---------+--------------+
|        | GET|HEAD | /         |      | Closure | web          |
|        | GET|HEAD | api/user  |      | Closure | api,auth:api |
|        | GET|HEAD | asdf      |      | Closure | web          |
|        | GET|HEAD | libro/{i} |      | Closure | web          |
+--------+----------+-----------+------+---------+--------------+
```

Laravel: Filtros

Filtrar rutas usando Middleware

Para capturar variables traspasadas en la ruta (GET) se debe usar el método $request->route()

- Ejemplo archivo web.php

```
Route::get('/clientes/{id}','Home')->middleware('myMiddleware');
```

- Implementación de Middleware

```
public function handle($request, Closure $next)
{
    if($request->route("client_id") == 0) return redirect('home');
    if(!(in_array("client_id",$request->route()->parameters())))
        return redirect("home");

    return $next($request); }
```