

Autenticar la API RESTful con un token por cabecera HTTP

Contenido

| | |
|---|---|
| Habilitar el middleware de autenticación..... | 2 |
| Revisar método handle del archivo Authenticate.php..... | 2 |
| Adaptar método boot de acuerdo a requerimientos de autenticación..... | 3 |
| Colocar el middleware de autenticación en las rutas en el archivo de rutas o en el controlador..... | 3 |
| Referencias web adicionales..... | 4 |



Habilitar el middleware de autenticación

Habilitar el middleware de autenticación y registrar el proveedor de autenticación en el archivo bootstrap/app.php

```
$app->routeMiddleware([
    'auth' => App\Http\Middleware\Authenticate::class,
]);

// Support for Authentication
$app->register(App\Providers\AuthServiceProvider::class);
```

Revisar método handle del archivo Authenticate.php

Tener presente el método handle en el archivo [App\Http\Middleware\Authenticate.php](#), este método se ejecutará antes de nuestros métodos protegidos. Así que cada vez que hacemos una solicitud para nuestros métodos de creación, actualización o eliminación, el middleware de autenticación se va a utilizar, y este método handle va a ser llamado.

Si el usuario no está autenticado, entonces devolverá un 401. De lo contrario, pasará la solicitud a la siguiente cosa que procesará esa solicitud.

```
public function handle($request, Closure $next, $guard = null)
{
    if ($this->auth->guard($guard)->guest()) {
        return response('Unauthorized.', 401);
    }
    return $next($request);
}
```



Adaptar método boot de acuerdo a requerimientos de autenticación

Revisar el archivo [/var/www/laravel/apirf.lumen/app/Providers/AuthServiceProvider.php](#) el método boot, este es el método que es responsable de autenticar al usuario. Así que de esto va a depender de nuestra implementación. Y en este ejercicio, nuestra implementación va a ser muy simple. Lo que vamos a hacer es buscar una cabecera llamada api-Token. Y si es un cierto valor, entonces vamos a decir que el usuario está autenticado. Para decir que un usuario es autenticado, tenemos que devolver una instancia de usuario. Si devolvemos null, entonces significa que el usuario no está autenticado.

```
$this->app['auth']->viaRequest('api', function ($request) {
    $api_token = $request->header('api-token'); // IMPORTANT: Name to header must be spinal-
case
    if ($api_token && $api_token=='freedom'){
        return new User(); // This is didactic
    }
    return null;
});
```

IMPORTANT: Name to header must be spinal-case

Colocar el middleware de autenticación en las rutas en el archivo de rutas o en el controlador

Colocar el middleware auth en las rutas en el archivo de rutas o como segunda opción es especificar nuestro middleware dentro del constructor de nuestro controlador. Eso depende de la necesidad

```
// Other way by each route, but I prefer in the Controller
// $router->get('manufacturers', ['middleware'=>'auth', 'uses' => 'ManufacturerController@index']);
$router->get('manufacturers', ['uses' => 'ManufacturerController@index']);
```



Repositorio: <https://github.com/riv4wi/apirf-laravel>

Rama: [ft-authentication-with-token-very-simple](#)

Nombre-documento: [ft-authentication-with-token-very-simple.odt](#)

```
public function __construct()
{
    $this->middleware('auth', ['only' => [
        'store',
        'update',
        'destroy'
    ]]);
}
```

Referencias web adicionales

<https://code.tutsplus.com/es/tutorials/how-to-secure-a-rest-api-with-lumen-%2Dcms-27442>



Repositorio: <https://github.com/riv4wi/apirf-laravel>

Rama: **ft-authentication-with-token-very-simple**

Nombre documento: **ft-authentication-with-token-very-simple.odt**