

Laravel Authentication Middleware

PHP WebDevelopment 2020-2021

Milena Tomova
Borislav Krustev
Vratsa Software

<https://vratsasoftware.com/>

Table of Contents



1. Frontend Scaffolding
2. Authentication
3. Middleware

Learn to Search in Internet

- The course assignments require to search in Internet
 - This is an important part of the learning process
 - Some exercises intentionally have no hints
- Learn to find solutions!
 - Software development includes everyday searching and learning
 - No excuses, just learn to study!
 - Developers learn new technologies, tools, languages every day!





Frontend Scaffolding

Frontend Scaffolding

Frontend Scaffolding

While Laravel does not dictate which JavaScript or CSS pre-processors you use, it does provide a basic starting point using [Bootstrap](#), [React](#), and / or [Vue](#) that will be helpful for many applications.

Frontend Scaffolding

- The **Bootstrap** and **Vue** are located in the **laravel/ui** Composer package
 - which may be installed by

```
composer require laravel/ui:^2.4
```

Frontend Scaffolding

- then you may install **the frontend scaffolding** of your choice using the ui Artisan command

```
// Generate basic scaffolding...
```

```
php artisan ui bootstrap
```

```
php artisan ui vue
```

```
php artisan ui react
```

- run **npm install** and **npm run dev**

Frontend Scaffolding

- to generate **Laravel Auth** scaffolding

```
// Generate login / registration scaffolding...  
php artisan ui bootstrap --auth  
php artisan ui vue --auth  
php artisan ui react --auth
```




Authentication

Authentication

Authentication

Laravel makes implementing authentication very simple.

In fact, almost everything is configured for you out of the box.

The authentication configuration file is located at `config/auth.php`, which contains several well documented options for tweaking the behavior of the authentication services.

- to generate **Laravel Auth** scaffolding

```
// Generate login / registration scaffolding...
```

```
php artisan ui bootstrap --auth
```

```
php artisan ui vue --auth
```

```
php artisan ui react --auth
```

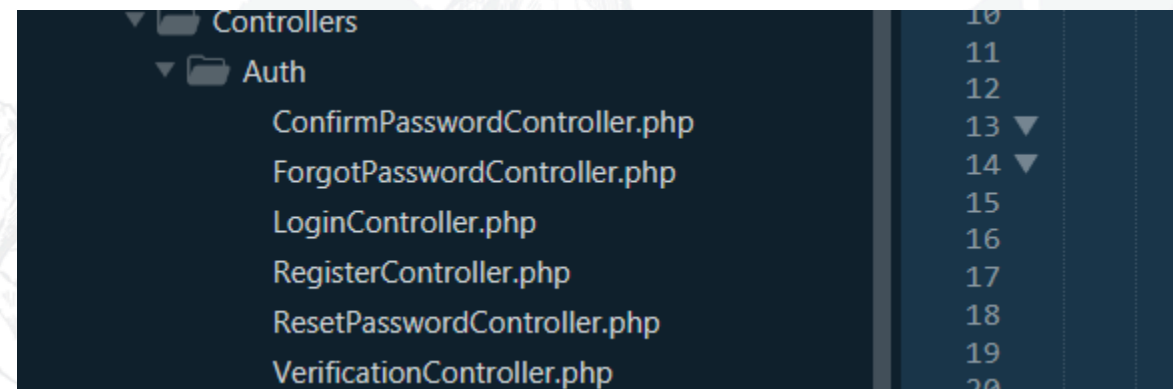

Authentication

This command creates -

- auth routes

```
18 }  
19 },  
20 Auth::routes();  
21
```

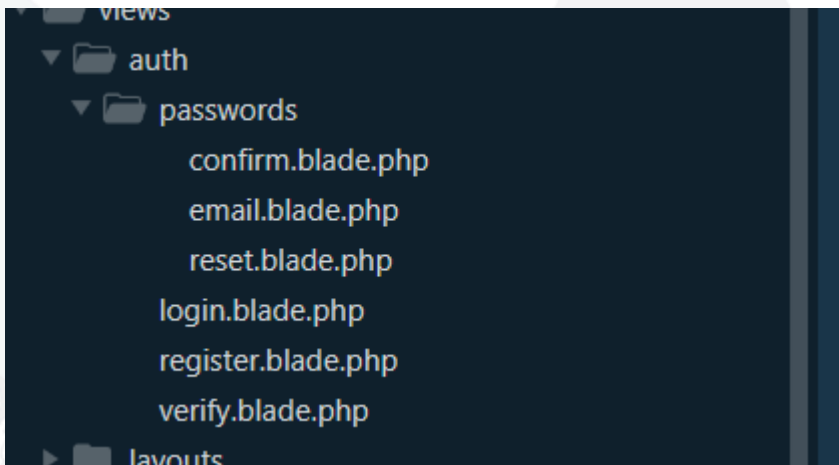
- auth controllers



Authentication

- auth views - backend

- frontend



ravel

LOGIN

REGISTER

OG

NOVA

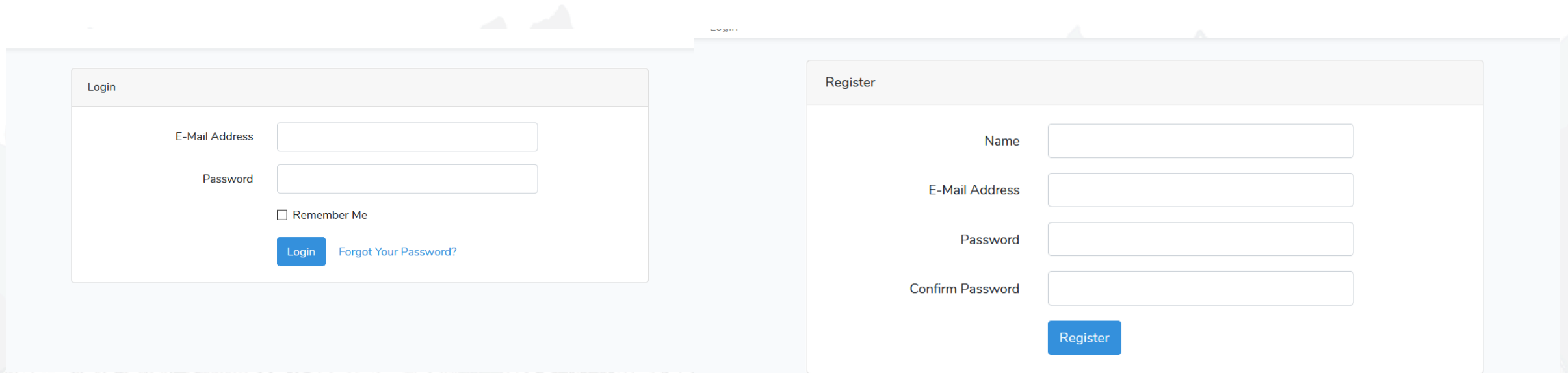
FORGE

VAPOR

GITHUB



- frontend auth views



The image displays two side-by-side form views for authentication. The left view is titled 'Login' and contains fields for 'E-Mail Address' and 'Password', a 'Remember Me' checkbox, a 'Login' button, and a 'Forgot Your Password?' link. The right view is titled 'Register' and contains fields for 'Name', 'E-Mail Address', 'Password', and 'Confirm Password', along with a 'Register' button. Both forms are presented in a clean, modern style with light gray borders and white backgrounds.

Login

E-Mail Address

Password

☐ Remember Me

[Login](#) [Forgot Your Password?](#)

Register

Name

E-Mail Address

Password

Confirm Password

[Register](#)



Authentication

Authentication

The auth system is fully customisable.

You can change -

- the user login/registration
- auth driver
- design

[auth customisations](#)

Customising auth logic flow

If you need to change customisation logic you **do not change** the methods in **AuthenticatesUsers** trait.

You redefine auth properties and methods in the appropriate controllers as shown in the [documentation](#)



Authentication

Authentication

Retrieving the authenticated user

You may access the authenticated user via the **Auth** facade - always place on top of the file, holding the class definition -

use Illuminate\Support\Facades\Auth;

- in the controller or model -

```
// Get the currently authenticated user...
```

```
$user = Auth::user();
```

```
// Get the currently authenticated user's ID...
```

```
$id = Auth::id();
```

- Once a user is authenticated, you may access the **authenticated user**
 - via an **Illuminate\Http\Request** instance.
 - Remember, type-hinted classes will automatically be injected into your controller methods:

```
public function update(Request $request)
{
    // $request->user() returns an instance of the authenticated user...
}
```


- To determine if the **user is already logged** into your application -

```
if( Auth::check() ) {  
    // The user is logged in..  
}
```


Determining If The Current User Is Authenticated



Even though it is possible to determine if a user is authenticated using the check method, you will typically **use a middleware to verify** that the user is authenticated before allowing **the user access to certain routes / controllers.**

- To allow only **authenticated users** to access a given route.

```
Route::get('profile', function () {  
    // Only authenticated users may enter...  
})->middleware('auth');
```


- Protect **all controller methods** by calling a middleware in controller`s constructor

```
public function __construct()  
{  
    $this->middleware('auth');  
}
```


Laravel ships with an **auth** middleware, which is defined at `Illuminate\Auth\Middleware\Authenticate`. Since this middleware is already registered in your HTTP kernel, all you need to do is attach the middleware to a route definition(s)

- Redirect **unauthenticated users to named route**

```
protected function redirectTo($request)
{
    return route('login');
}
```

modify this behavior by updating the **redirectTo** function in your **app/Http/Middleware/Authenticate.php** file

Authentication

Authentication

For more on Laravel authentication, please read the [documentation](#)



Middleware

Middleware

Middleware

- **Middleware** provide a convenient mechanism for filtering **HTTP requests** entering your application - **for protecting the routes of the application.**
- For example, Laravel includes a middleware that verifies **the user** of your application **is authenticated.**
- If the user is not authenticated, the middleware **will redirect the user to the login screen.**
- However, if the user is authenticated, the middleware **will allow the request to proceed** further into the application. [documentation](#)

Middleware

Middleware

Additional middleware can be written to perform a variety of tasks besides authentication.

Let`s build **a middleware** that will protect **the admin routes** - the routes for reading, adding, editing, deleting users.

- create a new middleware

```
php artisan make:middleware CheckIsAdmin
```

This command will place a new **CheckIsAdmin** class within your **app/Http/Middleware** directory.

- In this middleware, we will only allow access to the route if the user role is admin. Otherwise, we will redirect the users back to the home URI.

```
public function handle($request, Closure $next)
{
    if ($request->user()->role->name != 'admin') {
        return redirect('home');
    }
    return $next($request);
}
```


- The middleware runs **before** the request is handled by the app

```
public function handle($request, Closure $next)
{
    if ($request->user()->role->name != 'admin') {
        return redirect('home');
    }
    return $next($request);
}
```


- The middleware runs **after** the request is handled by the app

```
public function handle($request, Closure $next)
{
    $response = $next($request);
    // Perform middleware check action
    return $response;
}
```

- Register the middleware in `app/Http/Kernel.php` if you want to run during every HTTP request to your application in the **\$middleware** property.
- If you want to assign the middleware to specific routes
 - follow the steps in the [documentation](#).

- To assign **multiple** middlewares -

```
Route::resource('/demo', 'demoController')->middleware(['auth', 'admin']);
```

See [the documentation](#) for available ways of **grouping routes**.

Questions?



Гнездото
Coworking

Цялостен
курс по
програми
ране

Дизайн
курс

Курс по
дигит.
маркетинг

MindHub



Partners



**Telerik
Academy**



MindHub

ПРОМЯНАТА

Trainings @ Vratsa Software



- Vratsa Software – High-Quality Education, Profession and Jobs
 - www.vratsasoftware.com
- The Nest Coworking
 - www.gnezdoto.vratsasoftware.com
- Vratsa Software @ Facebook
 - www.fb.com/VratsaSoftware
- Slack Channel
 - www.vso.slack.com

