

Laravel Multi Auth

Use Role Type

1. First of all, install a fresh Laravel 8 (jetstream livewire or inertia).
2. Create a new column(role_id or role_type) in your Users table.

```
database > migrations > 2014_10_12_000000_create_users_table.php > ...
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateUsersTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->id();
18             $table->string('role_id')->default('1')->comment('1 => users, 2 => Admin');
19             $table->string('name');
20             $table->string('email')->unique();
21             $table->timestamp('email_verified_at')->nullable();
22             $table->string('password');
23             $table->rememberToken();
24             $table->foreignId('current_team_id')->nullable();
25             $table->string('profile_photo_path', 2048)->nullable();
26             $table->boolean('status')->default(1)->comment('0 => inActive, 1 => Active, 2 => freez, 3 => Block');
27             $table->timestamps();
28         });
29     }
30 }
```

3. Configure your (.env)file (Connection your database)

```
.env
1  APP_NAME="Multi Auth Jetstream"
2  APP_ENV=local
3  APP_KEY=base64:zrKgYlMLMxZwMFz68B747sq42SW86Yzrqet723FB1CE=
4  APP_DEBUG=true
5  APP_URL=http://localhost:8000
6
7  LOG_CHANNEL=stack
8  LOG_LEVEL=debug
9
10 DB_CONNECTION=mysql
11 DB_HOST=127.0.0.1
12 DB_PORT=3306
13 DB_DATABASE=multi_auth_jetstream
14 DB_USERNAME=root
15 DB_PASSWORD=
16
17 BROADCAST_DRIVER=log
18 CACHE_DRIVER=file
19 FILESYSTEM_DRIVER=local
20 QUEUE_CONNECTION=sync
21 SESSION_DRIVER=database
22 SESSION_LIFETIME=120
23
24 MEMCACHED_HOST=127.0.0.1
25
```

4. Migrate your database (Run "php artisan migrate")
5. Create middleware(Like User, Admin, Doctor, Patient, etc). It depends on your user type. (Run "php artisan make:middleware Admin")
6. Define your middleware in the (kernel.php) file. (App\Http\Kernel.php). Follow the image below.

```
app > Http > Kernel.php > ...
50
51 /**
52  * The application's route middleware.
53  *
54  * These middleware may be assigned to groups or used individually.
55  *
56  * @var array
57  */
58 protected $routeMiddleware = [
59     'auth' => \App\Http\Middleware\Authenticate::class,
60     'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
61     'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
62     'can' => \Illuminate\Auth\Middleware\Authorize::class,
63     'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
64     'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
65     'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
66     'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
67     'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
68
69     // Admin
70     'Admin' => \App\Http\Middleware\Admin\Admin::class,
71
72     // User
73     'User' => \App\Http\Middleware\User\User::class,
74 ];
75
76
```

7. Redirect your users on your middleware file. Follow the image below.
(App\Http\Middleware\Admin.php).

```
app > Http > Middleware > Admin > Admin.php > ...
1  <?php
2
3  namespace App\Http\Middleware\Admin;
4
5  use Closure;
6  use Facade\FlareClient\Http\Response;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Auth;
9  use Illuminate\Support\Facades\DB;
10
11 class Admin
12 {
13     /**
14      * Handle an incoming request.
15      *
16      * @param \Illuminate\Http\Request $request
17      * @param \Closure $next
18      * @return mixed
19      */
20     public function handle(Request $request, Closure $next)
21     {
22         if (!Auth::check()) {
23             return redirect()->route('admin.login');
24         }
25         $user = DB::table('users')->select('role_id')->where('id', Auth::user()->id)->first();
26         if ($user->role_id == 2) {
27             return $next($request);
28         }
29         return abort(403, 'Not authorized to access this page.');
```

8. Create a route ('/redirect') to redirect your users. Follow the image below.

```
routes > web.php > ...
1  <?php
2
3  use Illuminate\Foundation\Application;
4  use Illuminate\Support\Facades\Route;
5  use Inertia\Inertia;
6  use Illuminate\Support\Facades\Auth;
7
8  Route::get('/redirect', function () {
9      return Auth::user()->role_id == 2 ? redirect()->route('admin.dashboard') : redirect()->route('dashboard');
10 });
11
12
13 /*
14 |-----
15 | Web Routes
16 |-----
17
```

9. Define the route ('/redirect') in RouteServiceProvider.php file.
(App\Providers\RouteServiceProvider.php). Follow the image below.

```
app > Providers > RouteServiceProvider.php > ...
1  <?php
2
3  namespace App\Providers;
4
5  use Illuminate\Cache\RateLimiting\Limit;
6  use Illuminate\Foundation\Support\Providers\RouteServiceProvider as ServiceProvider;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\RateLimiter;
9  use Illuminate\Support\Facades\Route;
10
11 class RouteServiceProvider extends ServiceProvider
12 {
13     /**
14      * The path to the "home" route for your application.
15      *
16      * This is used by Laravel authentication to redirect users after login.
17      *
18      * @var string
19      */
20     public const HOME = '/redirect';
21
22
23
24     /**
25      * The controller namespace for the application.
26      *
27      * When present, controller route declarations will automatically be prefixed with this namespace.
28      *
29      * @var string|null
30      */
31     // protected $namespace = 'App\\Http\\Controllers';
32
33     /**
34      * Define your route model bindings, pattern filters, etc.
35      *
36      * @return void
37      */
38 }
```

10. Use middleware and prefix(use role type like "admin or user") and also use the group method to group all the routes in your single user. Follow the image below.

```

routes > web.php > ...
1  <?php
2
3  use Illuminate\Foundation\Application;
4  use Illuminate\Support\Facades\Route;
5  use Inertia\Inertia;
6  use Illuminate\Support\Facades\Auth;
7
8  Route::middleware(['Admin', 'auth:sanctum', 'verified'])->prefix('admin')->group(function () {
9      Route::get('/dashboard', function () {
10         return Inertia::render('Admin/Dashboard');
11     }->name('admin.dashboard'));
12 });
13 /*
14 |-----
15 | Web Routes
16 |-----

```

11. Redirect login page if users are not authenticated.
(App\Http\Middleware\Authenticate.php). Follow the image below.

```

app > Http > Middleware > Authenticate.php > ...
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Illuminate\Auth\Middleware\Authenticate as Middleware;
6
7  class Authenticate extends Middleware
8  {
9      /**
10       * Get the path the user should be redirected to when they are not authenticated.
11       *
12       * @param  \Illuminate\Http\Request  $request
13       * @return string|null
14       */
15      protected function redirectTo($request)
16      {
17          if (!$request->expectsJson()) {
18              // return route('login'); //use default code if you use a single login page
19              return $request->segment(1) == 'admin' ? route('admin.login') : route('user.login'); //use this code if you use a different login page
20          }
21      }
22  }
23
24

```

12. Redirect the default login page in your route file(web.php) if you want to use multiple login pages. Follow the image below.

```

routes > web.php > ...
1  <?php
2
3  use Illuminate\Foundation\Application;
4  use Illuminate\Support\Facades\Route;
5  use Inertia\Inertia;
6  use Illuminate\Support\Facades\Auth;
7
8  Route::get('/login', function () {
9      return redirect()->route('user.login');
10 }->name('login');
11
12 /*
13 |-----
14 | Web Routes
15 |-----

```

Author: Md Risul Islam Munna

Web: <https://risulislamunna.com>

