



Laravel 8 Multi Auth (Authentication) Tutorial

By Hardik Savani October 7, 2020 Category : Laravel

Powered By **VDO.AI**



In this example, you will learn laravel 8 multi auth. I would like to share with you laravel 8 multiple auth. This post will give you simple example of laravel 8 multiple authentication.

I explained simply step by step multiple authentication in laravel 8.

I written many tutorials about multi authentication in laravel. in this tutorial we will create multi auth very simple way using middleware with single table. if you want to create multiple authentication using guard than you can follow this tutorial: [Laravel multi auth example using Auth guard from scratch](#) and if you want to create multiple authentication with laravel using role and middleware than you can follow this tuto: [Laravel 5 - Simple user access control using Middleware](#)

However, in this example, we will create very simple way and you can easily use with your laravel 8 application. so let's follow this step.



MULTI AUTHENTICATION TUTORIAL

ITSOLUTIONSTUFF.COM

Step 1: Install Laravel 8

first of all we need to get fresh Laravel 8 version application using bellow command, So open your terminal OR command prompt and run bellow command:

```
composer create-project --prefer-dist laravel/laravel blog
```

Step 2: Database Configuration

In second step, we will make database configuration for example database name, username, password etc for our crud application of laravel 8. So let's open .env file and fill all details like as bellow:

.env

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=here your database name(blog)
DB_USERNAME=here database username(root)
DB_PASSWORD=here database password(root)
```

Read Also: [Laravel 8 Create Custom Helper Functions Tutorial](#)

Step 3: Update Migration and Model

In this step, we need to add new row "is_admin" in users table and model. than we need to run migration. so let's change that on both file.

database/migrations/000_create_users_table.php

```
<?php

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email');
            $table->timestamp('email_verified_at')->nullable();
            $table->boolean('is_admin')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```

app/Models/User.php

```
<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
```

```

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password', 'is_admin'
    ];

    /**
     * The attributes that should be hidden.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    /**
     * The attributes that should be cast to native types.
     *
     * @var array
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}

```

Now we need to run migration.

so let's run bellow command:

```
php artisan migrate
```

Step 4: Create Auth using scaffold

Now, in this step, we will create auth scaffold command to create login, register and dashboard. so run following commands:

Laravel 8 UI Package

```
composer require laravel/ui
```

Generate auth

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

```
npm install
```

```
npm run dev
```

Step 5: Create IsAdmin Middleware

In this step, we will create a middleware named `IsAdmin` which checks if the user is an admin or not.

↳ admin user

```
php artisan make:middleware IsAdmin
```

app/Http/middleware/IsAdmin.php

```
<?php

namespace App\Http\Middleware;

use Closure;

class IsAdmin
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if(auth()>user()>is_admin == 1){
            return $next($request);
        }

        return redirect('home')->with('error',"You don't have admin access.");
    }
}
```

app/Http/Kernel.php

```
.....
protected $routeMiddleware = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
    'is_admin' => \App\Http\Middleware\IsAdmin::class,
```

```
];
....
```

Step 6: Create Route

Here, we need to add one more route for admin user home page so let's add that route in web.php file.

routes/web.php

```
<?php

use Illuminate;
use App\Http;

/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/
Route::get('admin/home', [HomeController::class, 'adminHome'])->name('admin.home')->middleware('is_admin');
```

Step 7: Add Method on Controller

Here, we need add adminHome() method for admin route in HomeController. so let's add like as bellow:

app/Http/Controllers/HomeController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth');
    }

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Contracts\Support\Renderable
     */
    public function index()
```

```

    {
        return view('home');
    }

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Contracts\Support\Renderable
     */
    public function adminHome()
    {
        return view('adminHome');
    }
}

```

Step 8: Create Blade file

In this step, we need to create new blade file for admin and update for user blade file. so let's change it.

resources/views/home.blade.php

```

@extends('layouts.app')

@section('content')


Dashboard



You are normal user.


@endsection


```

resources/views/adminHome.blade.php

```

@extends('layouts.app')

@section('content')

```

```

<div class="col-md-8">
    <div class="card">
        <div class="card-header">Dashboard</div>
        <div class="card-body">
            You are Admin.
        </div>
    </div>
</div>
@endsection

```

Step 9: L

In this step, we will change on LoginController, when user will login than we redirect according to user access. If normal user than we will redirect to home route and if admin user than we redirect to admin route. so let's change.

app/Http/Controllers/Auth/LoginController.php

```

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\AuthenticatesUsers;
use Illuminate\Http\Request;

class LoginController extends Controller
{
    /*
    |--------------------------------------------------------------------------
    | Login Controller
    |--------------------------------------------------------------------------
    |
    | This controller handles authenticating users for the application and
    | redirecting them to your home screen. The controller uses a trait
    | to conveniently provide its functionality to your applications.
    |
    */

    use AuthenticatesUsers;

    /**
     * Where to redirect users after login.
     *
     * @var string
     */
    protected $redirectTo = '/home';

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest')->except('logout');
    }

    public function login(Request $request)

```

```

{
    $input = $request->all();

    $this->validate($request, [
        'email' => 'required|email',
        'password' => 'required',
    ]);

    if(auth()->attempt(array('email' => $input['email'], 'password' => $input['password'])))
    {
        if (auth()->user()->is_admin == 1) {
            return redirect()->route('admin.home');
        }else{
            // User is not Admin
        }
    }
}

```

Step 10: Create Seeder

We will create seeder for create new admin and normal user. so let's create seeder using following command:

```
php artisan make:seeder CreateUsersSeeder
```

database/seeds/CreateUsersSeeder.php

```

<?php

use Illuminate\Database\Seeder;
use App\Models\User;

class CreateUsersSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $user = [
            [
                'name'=>'Admin',
                'email'=>'admin@itsolutionstuff.com',
                'is_admin'=>'1',
                'password'=> bcrypt('123456'),
            ],
            [
                'name'=>'User',
                'email'=>'user@itsolutionstuff.com',
                'is_admin'=>'0',
                'password'=> bcrypt('123456'),
            ],
        ];
    }
}

```

```

        foreach ($user as $key => $value) {
            User::create($value);
        }
    }
}

```

Now let's run seeder:

```
php artisan db:seed --class=CreateUsersSeeder
```

Ok, now we are ready to run.

So let's run pro

```
php artisan s
```

Admin User

Email: admin@itsolutionstuff.com
Password: 123456

Normal User

Read Also: [Laravel 8 Install React Example Tutorial](#)

Email: user@itsolutionstuff.com
Password: 123456

You can download code from git: [Download Code from Github](#)

I hope it can help you...

Tags : Auth Laravel Laravel 8



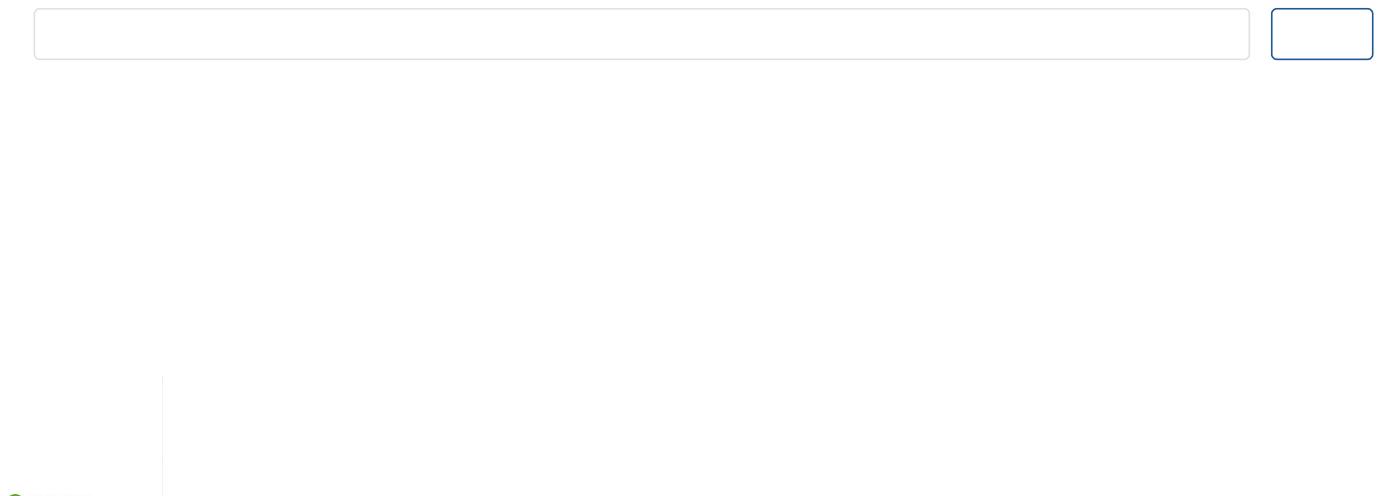
Hardik Savani

I'm a full-stack developer, entrepreneur and owner of Aatman Infotech. I live in India and I love to write tutorials and tips that can help to other art

y stage. I

We are Recommending you

- › [Laravel 8 Install Bootstrap Example Tutorial](#)
- › [Laravel 8 Markdown | Laravel 8 Send Email using Markdown Example](#)
- › [Drag & Drop File Uploading using Laravel 8 Dropzone JS](#)
- › [Laravel 8 REST API with Passport Authentication Tutorial](#)
- › [Laravel 8 Socialite Login with Google Account Example](#)
- › [Laravel 8 Send Mail using Queue Example](#)
- › [Laravel 8 Import Export Excel and CSV File Tutorial](#)
- › [Laravel 8 Guzzle Http Client Request Example](#)
- › [Laravel 8 Livewire CRUD with Jetstream & Tailwind CSS](#)
- › [Laravel 8 PDF | Laravel 8 Generate PDF File using DomPDF](#)
- › [Laravel 8 Create Custom Helper Functions Tutorial](#)



ezoic

Popular Posts

- › [Laravel - Line Chart Example using Google Charts API](#)
- › [Laravel 9 Socialite Login with Github Account Example](#)
- › [Laravel 9 Google Recaptcha V3 Example Tutorial](#)
- › [Laravel 9 Yajra Datatables Example Tutorial](#)
- › [Laravel 5 - Simple CRUD Application Using ReactJS - Part 1](#)
- › [How to solve - This cache store does not support tagging?](#)
- › [Laravel - How to remove foreign key constraint using migration?](#)
- › [Laravel Generate Sitemap using roumen/sitemap package Example](#)
- › [How to Compare Two Dates in Laravel Carbon?](#)
- › [Laravel Blade @includeWhen and @includeUnless Example](#)
- › [How to Generate BarCode in Laravel?](#)

Categories

Laravel

Angular

PHP

jQuery

Python

Bootstrap

Javascript

MySQL

Ajax

Node JS

HTML

Ubuntu

Codeigniter

Guest Post

Vue.JS

React JS

Git

Server

[JSON](#)[Installation](#)[CSS](#)[Google Map](#)[SQL](#)[.htaccess](#)[Google API](#)[JQuery UI](#)[Typeahead JS](#)[Axios](#)[Socket.io](#)[Elasticsearch](#)

Latest Posts

- [How to Refresh Laravel Session in Controller](#)
- [Laravel TC](#)
- [Laravel Delete File After Download Response Example](#)
- [How to Take Browser Screenshots in Laravel?](#)
- [Laravel Carbon Count Weekends Days Between Two Dates Example](#)

[report this ad](#)

Subscribe to the Email Newsletter

Enter Your Email Address....



Follow Laravel News on

[Facebook](#)

[Twitter](#)

[LinkedIn](#)

[Github](#)

[Home](#)

[List Of Categories](#)

[List of Tags](#)

[Disclaimer](#)

[Latest Posts](#)

[Contact US](#)

[About US](#)

[Advertisement](#)

© 2016-2022 All Rights Reserved • www.itsolutionstuff.com

