

# Laravel 8 User Roles and Permissions Tutorial

Shares

f

👤 By Hardik Savani 📅 November 5, 2023 🗂️ Category : Laravel

X

Today our leading topic is laravel 8 roles and permissions tutorial. In this article, we will implement a laravel 8 spatie user roles and permissions tutorial. i explained simply step by step laravel 8 spatie/laravel-permission. step by step explain laravel 8 acl tutorial.

p

in

✉



we are using spatie github package for roles and permissions in laravel 8 application. just follow bellow step to create acl in laravel 8.

Spatie role permission composer package provide way to create acl in laravel 8. they provide how to assign role to user, how to assign permission to user and how to assign permission assign to roles. i will write step by step creating

roles and permissions in laravel 8 application.

Shares



I also posted on tutorial for ACL User Roles and Permissions using entrust package, you can see here : [Laravel - User Roles and Permissions \(ACL\) using entrust package](#).

Roles and Permissions through you can create several types of users with different role and permission, i mean some user have only see listing of items module, some user can also edit items modules, for delete and etc.

In this examples i created three modules as listed bellow:

User Management

Role Management

## Product Management

After register user, you don't have any roles, so you can edit your details and assign admin role to you from User Management. After that you can create your own role with permission like role-list, role-create, role-edit, role-delete, product-list, product-create, product-edit, product-delete. you can check with assign new user and check that.

Shares



You need to just follow few step and you will get full example of ACL:

Laravel 8 User Roles and Permissions - ItSolutionStuff.com

Manage Users Manage Role Manage Product Hardik Savani ▾

### Users Management

Create New User

No	Name	Email	Roles	Action
1	Harshad Pathak	itsolutionstuff@gmail.com	User	Show Edit Delete
2	Paresh Patel	aatmaninfotech@gmail.com	User	Show Edit Delete
3	Hardik Savani	admin@gmail.com	Admin	Show Edit Delete

Tutorial by ItSolutionStuff.com

**List Role:**

**Role Management**[Create New Role](#)

No	Name	Action
1	User	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a>
2	Admin	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Delete</a>

Tutorial by ItsolutionStuff.com

Shares

**Create Role:**

Shares



Laravel 8 User Roles and Permissions - ItSolutionStuff.com

Manage Users | Manage Role | Manage Product | Hardik Savani ▾

### Create New Role

[Back](#)

Name:

Permission:

role-list  
 role-create  
 role-edit  
 role-delete  
 product-list  
 product-create  
 product-edit  
 product-delete

[Submit](#)

Tutorial by ItSolutionStuff.com

## Create User:

Laravel 8 User Roles and Permissions - ItSolutionStuff.com

Manage Users | Manage Role | Manage Product | Hardik Savani ▾

### Create New User

[Back](#)

Name:

Email:

Password:

Confirm Password:

Role:

[Submit](#)

Tutorial by ItSolutionStuff.com

## List Product:

## Products

[Create New Product](#)

Product created successfully.

No	Name	Details	Action
1	Silver	this is silver	Show Edit Delete
2	Gold	this is gold	Show Edit Delete

Tutorial by ItSolutionStuff.com

Shares



Step 1: Laravel 8 Installation

We are going from scratch so, If you haven't installed laravel in your system then you can run bellow command and get fresh Laravel project.

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

Shares



## Step 2: Install Composer Packages

Now we require to install Spatie package for ACL, that way we can use it's method. Also we will install form collection package. So Open your terminal and run bellow command.

```
composer require spatie/laravel-permission
```



```
composer require laravelcollective/html
```

Now open config/app.php file and add service provider and aliase.

## config/app.php

Shares



```
'providers' => [
    ...
    Spatie\Permission\PermissionServiceProvider::class,
],
```

We can also custom changes on Spatie package, so if you also want to changes then you can fire bellow command and get config file in config/permission.php and migration files.

```
php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider"
```



Now you can see permission.php file and one migrations. so you can run migration using following command:

```
php artisan migrate
```

Shares



Read Also: [Laravel 8 CRUD Application Tutorial for Beginners](#)



### Step 3: Create Product Migration



In this step we have to create three migration for products table as using bellow command:

```
php artisan make:migration create_products_table
```

## products table:

Shares

```
<?php

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

class CreateProductsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('products', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->text('detail');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
```



```
{  
    Schema::dropIfExists('products');  
}  
}
```

Now run migration:

Shares



```
php artisan migrate
```



Step 4: Create Models

In this step we have to create model for User and Product table, so if you get fresh project then you have User Model have so just replace code and other you should create.

## 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;
use Spatie\Permission\Traits\HasRoles;

class User extends Authenticatable
{
    use HasFactory, Notifiable, HasRoles;

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

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
```

Shares  
f  
X  
p  
in  
✉

```
        'password',
        'remember_token',
    ];

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

Shares

f

X

P

in

✉

## app/Models/Product.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    use HasFactory;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'detail'
    ];
}
```



Shares



## Step 5: Add Middleware

---

Shares

f

x

p

in

✉



Shares



Spatie package provide it's in-built middleware that way we can use it simply and that is display as bellow:

**role**

**permission**

So, we have to add middleware in Kernel.php file this way :

**app/Http/Kernel.php**

```
....  
protected $routeMiddleware = [  
    ....  
    'role' => \Spatie\Permission\Middleware\RoleMiddleware::class,
```

```
'permission' => \Spatie\Permission\Middlewares\PermissionMiddleware::class,  
'role_or_permission' => \Spatie\Permission\Middlewares\RoleOrPermissionMiddleware::class,  
]  
....
```

## Step 6: Create Authentication

You have to follow few step to make auth in your laravel 8 application.

First you need to install laravel/ui package as like bellow:

```
composer require laravel/ui
```

Here, we need to generate auth scaffolding in laravel 8 using laravel ui command. so, let's generate it by bellow command:

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

Now you need to run npm command, otherwise you can not see better layout of login and register page.

### Install NPM:

```
npm install
```

### Run NPM:

```
npm run dev
```

## Step 7: Create Routes

We require to add number of route for users module, products module and roles module. In this this route i also use middleware with permission for roles and products route, so add route this way:

Shares



### routes/web.php

```
<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\HomeController;
use App\Http\Controllers\RoleController;
use App\Http\Controllers\UserController;
use App\Http\Controllers\ProductController;

/*
|--------------------------------------------------------------------------
| 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('/', function () {
    return view('welcome');
});

Auth::routes();

Route::get('/home', [HomeController::class, 'index'])->name('home');
```

```
Route::group(['middleware' => ['auth']], function() {
    Route::resource('roles', RoleController::class);
    Route::resource('users', UserController::class);
    Route::resource('products', ProductController::class);
});
```

Shares



## Step 8: Add Controllers

In this step we have add three controller for users module, products module and roles module so you can create three controller like as bellow:



## app/Http/Controllers/UserController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Models\User;
use Spatie\Permission\Models\Role;
use DB;
use Hash;
use Illuminate\Support\Arr;

class UserController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index(Request $request)
    {
        $data = User::orderBy('id', 'DESC')->paginate(5);
        return view('users.index', compact('data'))
            ->with('i', ($request->input('page', 1) - 1) * 5);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        $roles = Role::pluck('name', 'name')->all();
```

Shares



```
        return view('users.create', compact('roles'));
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $this->validate($request, [
            'name' => 'required',
            'email' => 'required|email|unique:users,email',
            'password' => 'required|same:confirm-password',
            'roles' => 'required'
        ]);

        $input = $request->all();
        $input['password'] = Hash::make($input['password']);

        $user = User::create($input);
        $user->assignRole($request->input('roles'));

        return redirect()->route('users.index')
            ->with('success', 'User created successfully');
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        $user = User::find($id);
```

Shares



```
        return view('users.show', compact('user'));
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $user = User::find($id);
        $roles = Role::pluck('name', 'name')->all();
        $userRole = $user->roles->pluck('name', 'name')->all();

        return view('users.edit', compact('user', 'roles', 'userRole'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
        $this->validate($request, [
            'name' => 'required',
            'email' => 'required|email|unique:users,email,'.$id,
            'password' => 'same:confirm-password',
            'roles' => 'required'
        ]);

        $input = $request->all();
        if(!empty($input['password'])) {
            $input['password'] = Hash::make($input['password']);
        }
    }
}
```

Shares



```
    }else{
        $input = Arr::except($input, array('password'));
    }

    $user = User::find($id);
    $user->update($input);
    DB::table('model_has_roles')->where('model_id', $id)->delete();

    $user->assignRole($request->input('roles'));

    return redirect()->route('users.index')
        ->with('success', 'User updated successfully');
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    User::find($id)->delete();
    return redirect()->route('users.index')
        ->with('success', 'User deleted successfully');
}
```

Shares



## app/Http/Controllers/ProductController.php

---

Shares

f

X

P

in

✉



```
<?php

namespace App\Http\Controllers;

use App\Models\Product;
use Illuminate\Http\Request;

class ProductController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    function __construct()
```

```
{  
    $this->middleware('permission:product-list|product-create|product-edit|product-delete');  
    $this->middleware('permission:product-create', ['only' => ['create', 'store']]);  
    $this->middleware('permission:product-edit', ['only' => ['edit', 'update']]);  
    $this->middleware('permission:product-delete', ['only' => ['destroy']]);
}  
/**/  
 * Display a listing of the resource.  
 *  
 * @return \Illuminate\Http\Response  
 */  
public function index()  
{  
    $products = Product::latest()->paginate(5);  
    return view('products.index', compact('products'))  
        ->with('i', (request()->input('page', 1) - 1) * 5);
}  
  
/**/  
 * Show the form for creating a new resource.  
 *  
 * @return \Illuminate\Http\Response  
 */  
public function create()  
{  
    return view('products.create');
}  
  
/**/  
 * Store a newly created resource in storage.  
 *  
 * @param \Illuminate\Http\Request $request  
 * @return \Illuminate\Http\Response  
 */  
public function store(Request $request)  
{  
    request()->validate([

```

Shares



```
        'name' => 'required',
        'detail' => 'required',
    ]) ;

    Product::create($request->all());

    return redirect()->route('products.index')
        ->with('success', 'Product created successfully.');
}

/**
 * Display the specified resource.
 *
 * @param \App\Product $product
 * @return \Illuminate\Http\Response
 */
public function show(Product $product)
{
    return view('products.show', compact('product'));
}

/**
 * Show the form for editing the specified resource.
 *
 * @param \App\Product $product
 * @return \Illuminate\Http\Response
 */
public function edit(Product $product)
{
    return view('products.edit', compact('product'));
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Product $product

```

Shares



```
* @return \Illuminate\Http\Response
*/
public function update(Request $request, Product $product)
{
    $request->validate([
        'name' => 'required',
        'detail' => 'required',
    ]);

    $product->update($request->all());

    return redirect()->route('products.index')
        ->with('success', 'Product updated successfully');
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Product $product
 * @return \Illuminate\Http\Response
 */
public function destroy(Product $product)
{
    $product->delete();

    return redirect()->route('products.index')
        ->with('success', 'Product deleted successfully');
}
```

Shares



app/Http/Controllers/RoleController.php

---

Shares

f

X

P

in

✉



```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Spatie\Permission\Models\Role;
use Spatie\Permission\Models\Permission;
use DB;

class RoleController extends Controller
{
    /**
     * Display a listing of the resource.
     *

```

```
* @return \Illuminate\Http\Response
*/
function __construct()
{
    $this->middleware('permission:role-list|role-create|role-edit|role-delete', ['only' =>
    $this->middleware('permission:role-create', ['only' => ['create','store']]);
    $this->middleware('permission:role-edit', ['only' => ['edit','update']]);
    $this->middleware('permission:role-delete', ['only' => ['destroy']]);
}

/**
 * Display a listing of the resource.
 *
 * @return \Illuminate\Http\Response
 */
public function index(Request $request)
{
    $roles = Role::orderBy('id','DESC')->paginate(5);
    return view('roles.index',compact('roles'))
        ->with('i', ($request->input('page', 1) - 1) * 5);
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    $permission = Permission::get();
    return view('roles.create',compact('permission'));
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request

```

Shares



```
* @return \Illuminate\Http\Response
*/
public function store(Request $request)
{
    $this->validate($request, [
        'name' => 'required|unique:roles,name',
        'permission' => 'required',
    ]);

    $role = Role::create(['name' => $request->input('name')]);
    $role->syncPermissions($request->input('permission'));

    return redirect()->route('roles.index')
        ->with('success','Role created successfully');
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    $role = Role::find($id);
    $rolePermissions = Permission::join("role_has_permissions","role_has_permissions.permissions_id", "role_has_permissions.role_id", $id)
        ->get();

    return view('roles.show',compact('role','rolePermissions'));
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */

```

Shares



```
public function edit($id)
{
    $role = Role::find($id);
    $permission = Permission::get();
    $rolePermissions = DB::table("role_has_permissions")->where("role_has_permissions.role_id", $role->id)->pluck('role_has_permissions.permission_id', 'role_has_permissions.permission_id');
    $rolePermissions = $rolePermissions->all();

Shares
f
X
p
in
✉
```

```
    return view('roles.edit', compact('role', 'permission', 'rolePermissions'));
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    $this->validate($request, [
        'name' => 'required',
        'permission' => 'required',
    ]);

    $role = Role::find($id);
    $role->name = $request->input('name');
    $role->save();

    $role->syncPermissions($request->input('permission'));

    return redirect()->route('roles.index')
        ->with('success', 'Role updated successfully');
}

/**
 * Remove the specified resource from storage.
 *
```

```
* @param int $id
* @return \Illuminate\Http\Response
*/
public function destroy($id)
{
    DB::table("roles")->where('id',$id)->delete();
    return redirect()->route('roles.index')
        ->with('success', 'Role deleted successfully');
}
```

Shares

f

X

P

in

✉

## Step 9: Add Blade Files

In this step, we need to create following files as like listed bellow:



## Theme Layout

app.blade.php

## Users Module

Shares

f index.blade.php create.blade.php edit.blade.php show.blade.php

## Roles Module

X index.blade.php create.blade.php edit.blade.php show.blade.php

## Product Module

in index.blade.php create.blade.php edit.blade.php show.blade.php

✉ So, let's create following files:

### resources/views/layouts/app.blade.php

```
<html lang="{!! app() ->getLocale() !!}">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- CSRF Token -->
    <meta name="csrf-token" content="{!! csrf_token() !!}">
    <title>{!! config('app.name', 'Laravel 8 User Roles and Permissions Tutorial') !!}</title>
    <!-- Scripts -->
    <script src="{!! asset('js/app.js') !!}" defer></script>
    <!-- Fonts -->
    <link rel="dns-prefetch" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css?family=Raleway:300,400,600" rel="stylesheet" t
```

```
<!-- Styles -->
<link href="{{ asset('css/app.css') }}" rel="stylesheet">
</head>
<body>
    <div id="app">
        <nav class="navbar navbar-expand-md navbar-light navbar-laravel">
            <div class="container">
                <a class="navbar-brand" href="{{ url('/') }}>
                    Laravel 8 User Roles and Permissions - ItSolutionStuff.com
                </a>
                <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
                    <span class="navbar-toggler-icon"></span>
                </button>

                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <!-- Left Side Of Navbar -->
                    <ul class="navbar-nav mr-auto"></ul>

                    <!-- Right Side Of Navbar -->
                    <ul class="navbar-nav ml-auto">
                        <!-- Authentication Links -->
                        @guest
                            <li><a class="nav-link" href="{{ route('login') }}>{{ __('Login') }}</a>
                            <li><a class="nav-link" href="{{ route('register') }}>{{ __('Register') }}</a>
                        @else
                            <li><a class="nav-link" href="{{ route('users.index') }}>Manage Users
                            <li><a class="nav-link" href="{{ route('roles.index') }}>Manage Roles
                            <li><a class="nav-link" href="{{ route('products.index') }}>Manage Products
                            <li class="nav-item dropdown">
                                <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" data-bs-toggle="dropdown" data-bs-display="static">
                                    {{ Auth::user()->name }} <span class="caret"></span>
                                </a>

                                <div class="dropdown-menu" aria-labelledby="navbarDropdown">
                                    <a class="dropdown-item" href="{{ route('logout') }}"
                                        onclick="event.preventDefault();
                                                    document.getElementById('logout-form').submit();">{{ __('Logout') }}</a>
                                </div>
                            </li>
                        @endif
                    </ul>
                </div>
            </div>
        </nav>
    </div>
</body>
```

Shares



```
        {{ __('Logout') }}  
    </a>  
  
    <form id="logout-form" action="{{ route('logout') }}" method="post">  
        @csrf  
    </form>  
    </div>  
    </li>  
@endguest  
    </ul>  
    </div>  
    </div>  
</nav>  
  
<main class="py-4">  
    <div class="container">  
        @yield('content')  
    </div>  
    </main>  
</div>  
</body>  
</html>
```

**resources/views/users/index.blade.php**

Shares



---

Shares

f

X

P

in

✉



```
@extends('layouts.app')

@section('content')


<div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2>Users Management</h2>
        </div>
        <div class="pull-right">
            <a class="btn btn-success" href="{{ route('users.create') }}"> Create New User</a>
        </div>
    </div>
</div>

@if ($message = Session::get('success'))


```

```
<div class="alert alert-success">
    <p>{{ $message }}</p>
</div>
@endif



| No          | Name               | Email               | Roles                                                                                                                                                               | Action                                                                                                                                                                                                                                                                                                                                                         |
|-------------|--------------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {{ ++\$i }} | {{ \$user->name }} | {{ \$user->email }} | @if(!empty(\$user->getRoleNames()))     @foreach(\$user->getRoleNames() as \$v)         <label class="badge badge-success">{{ \$v }}</label>     @endforeach @endif | <a class="btn btn-info" href="{{ route('users.show', \$user-&gt;id) }}">Show</a> <a class="btn btn-primary" href="{{ route('users.edit', \$user-&gt;id) }}">Edit</a> {!! Form::open(['method' => 'DELETE', 'route' => ['users.destroy', \$user->id], 'style'=>''])     {!! Form::submit('Delete', ['class' => 'btn btn-danger']) !!}     {!! Form::close() !!} |



{!! $data->render() !!}
```

Shares



```
<p class="text-center text-primary"><small>Tutorial by ItSolutionStuff.com</small></p>
@endsection
```

## resources/views/users/create.blade.php

Shares



```
@extends('layouts.app')

@section('content')


<div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2>Create New User</h2>
        </div>


```

```
        <div class="pull-right">
            <a class="btn btn-primary" href="{{ route('users.index') }}> Back</a>
        </div>
    </div>

@if (count($errors) > 0)
    <div class="alert alert-danger">
        <strong>Whoops!</strong> There were some problems with your input.<br><br>
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif

{!! Form::open(array('route' => 'users.store', 'method'=>'POST')) !!}
<div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Name:</strong>
            {!! Form::text('name', null, array('placeholder' => 'Name','class' => 'form-control')) !!}
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Email:</strong>
            {!! Form::text('email', null, array('placeholder' => 'Email','class' => 'form-control')) !!}
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Password:</strong>
            {!! Form::password('password', array('placeholder' => 'Password','class' => 'form-control')) !!}
        </div>
    </div>
```

Shares



```
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
    <div class="form-group">
        <strong>Confirm Password:</strong>
        {!! Form::password('confirm-password', array('placeholder' => 'Confirm Password', 'c
    </div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
    <div class="form-group">
        <strong>Role:</strong>
        {!! Form::select('roles[]', $roles, [], array('class' => 'form-control', 'multiple'))
    </div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12 text-center">
    <button type="submit" class="btn btn-primary">Submit</button>
</div>
</div>
{!! Form::close() !!}

<p class="text-center text-primary"><small>Tutorial by Itsolutionstuff.com</small></p>
@endsection
```

**resources/views/users/edit.blade.php**

Shares



---

Shares

f

X

P

in

✉



```
@extends('layouts.app')

@section('content')


<div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2>Edit New User</h2>
        </div>
        <div class="pull-right">
            <a class="btn btn-primary" href="{{ route('users.index') }}"> Back</a>
        </div>
    </div>
</div>

@if (count($errors) > 0)


```

```
<div class="alert alert-danger">
    <strong>Whoops!</strong> There were some problems with your input.<br><br>
    <ul>
        @foreach ($errors->all() as $error)
            <li>{{ $error }}</li>
        @endforeach
    </ul>
</div>
@endif

{!! Form::model($user, ['method' => 'PATCH', 'route' => ['users.update', $user->id]]) !!}

<div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Name:</strong>
            {!! Form::text('name', null, array('placeholder' => 'Name', 'class' => 'form-control')) !!}
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Email:</strong>
            {!! Form::text('email', null, array('placeholder' => 'Email', 'class' => 'form-control')) !!}
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Password:</strong>
            {!! Form::password('password', array('placeholder' => 'Password', 'class' => 'form-control')) !!}
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Confirm Password:</strong>
            {!! Form::password('confirm-password', array('placeholder' => 'Confirm Password', 'class' => 'form-control')) !!}
        </div>
    </div>
</div>
```

Shares



```
<div class="form-group">
    <strong>Role:</strong>
    {!! Form::select('roles[]', $roles,$userRole, array('class' => 'form-control','mult
    </div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12 text-center">
    <button type="submit" class="btn btn-primary">Submit</button>
</div>
</div>
{!! Form::close() !!}

<p class="text-center text-primary"><small>Tutorial by Itsolutionstuff.com</small></p>
@endsection
```

Shares



**resources/views/users/show.blade.php**



```
@extends('layouts.app')

@section('content')


<div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2> Show User</h2>
        </div>
        <div class="pull-right">
            <a class="btn btn-primary" href="{{ route('users.index') }}"> Back</a>
        </div>
    </div>
</div>

<div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Name:</strong>
            {{ $user->name }}
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Email:</strong>
            {{ $user->email }}
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Roles:</strong>
            @if(!empty($user->getRoleNames()))
                @foreach($user->getRoleNames() as $v)
                    <label class="badge badge-success">{{ $v }}</label>
                @endforeach
            @endif
        </div>
    </div>


```

Shares



```
        </div>
    </div>
</div>
@endsection
```

## resources/views/roles/index.blade.php

Shares



```
@extends('layouts.app')

@section('content')


## Role Management


```

```
<div class="pull-right">
@can('role-create')
    <a class="btn btn-success" href="{{ route('roles.create') }}"> Create New Role</a>
@endcan
</div>
</div>
</div>
```

Shares



```
@if ($message = Session::get('success'))
<div class="alert alert-success">
    <p>{{ $message }}</p>
</div>
@endif

<table class="table table-bordered">
<tr>
    <th>No</th>
    <th>Name</th>
    <th width="280px">Action</th>
</tr>
@foreach ($roles as $key => $role)
<tr>
    <td>{{ ++$i }}</td>
    <td>{{ $role->name }}</td>
    <td>
        <a class="btn btn-info" href="{{ route('roles.show', $role->id) }}">Show</a>
        @can('role-edit')
            <a class="btn btn-primary" href="{{ route('roles.edit', $role->id) }}">Edit</a>
        @endcan
        @can('role-delete')
            {!! Form::open(['method' => 'DELETE', 'route' => ['roles.destroy', $role->id], 's
                !!! Form::submit('Delete', ['class' => 'btn btn-danger']) !!!
           >{!! Form::close() !!}
        @endcan
    </td>
</tr>
@endforeach
```

```
</table>

{!! $roles->render() !!}



<small>Tutorial by ItSolutionStuff.com</small></p>
@endsection


```

Shares



[resources/views/roles/create.blade.php](#)



```
@extends('layouts.app')

@section('content')

```

```
<div class="pull-left">
    <h2>Create New Role</h2>
</div>
<div class="pull-right">
    <a class="btn btn-primary" href="{{ route('roles.index') }}"> Back</a>
</div>
</div>
</div>
```

Shares

f

X

P

in

✉

```
@if (count($errors) > 0)
<div class="alert alert-danger">
    <strong>Whoops!</strong> There were some problems with your input.<br><br>
    <ul>
        @foreach ($errors->all() as $error)
            <li>{{ $error }}</li>
        @endforeach
    </ul>
</div>
@endif

{!! Form::open(array('route' => 'roles.store','method'=>'POST')) !!}
<div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Name:</strong>
            {!! Form::text('name', null, array('placeholder' => 'Name','class' => 'form-control')) !!}
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Permission:</strong>
            <br/>
            @foreach($permission as $value)
                <label>{{ Form::checkbox('permission[]', $value->id, false, array('class' => 'n
                    {{ $value->name }})</label>
                <br/>
            @endforeach
        </div>
    </div>
</div>
```

```
        </div>
    </div>
<div class="col-xs-12 col-sm-12 col-md-12 text-center">
    <button type="submit" class="btn btn-primary">Submit</button>
</div>
</div>
{!! Form::close() !!}

<p class="text-center text-primary"><small>Tutorial by ItSolutionStuff.com</small></p>
@endsection
```

Shares



## resources/views/roles/edit.blade.php

```
@extends('layouts.app')
```



```
@section('content')


<div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2>Edit Role</h2>
        </div>
        <div class="pull-right">
            <a class="btn btn-primary" href="{{ route('roles.index') }}"> Back</a>
        </div>
    </div>


```

Shares

f

X

P

in

✉

```
@if (count($errors) > 0)
    <div class="alert alert-danger">
        <strong>Whoops!</strong> There were some problems with your input.<br><br>
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif

{!! Form::model($role, ['method' => 'PATCH', 'route' => ['roles.update', $role->id]]) !!}
<div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Name:</strong>
            {!! Form::text('name', null, array('placeholder' => 'Name', 'class' => 'form-control')) !!}
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Permission:</strong>
            <br/>
            @foreach ($permission as $value)

```

```
    <label>{{ Form::checkbox('permission[]', $value->id, in_array($value->id, $role  
    { { $value->name } })}</label>  
    <br/>  
    @foreach  
    </div>  
    </div>  
    <div class="col-xs-12 col-sm-12 col-md-12 text-center">  
        <button type="submit" class="btn btn-primary">Submit</button>  
    </div>  
    </div>  
{!! Form::close() !!}  
  
@endsection  
<p class="text-center text-primary"><small>Tutorial by Itsolutionstuff.com</small></p>
```

Shares



## resources/views/roles/show.blade.php

```
@extends('layouts.app')  
  
@section('content')  
<div class="row">  
    <div class="col-lg-12 margin-tb">  
        <div class="pull-left">  
            <h2> Show Role</h2>  
        </div>  
        <div class="pull-right">  
            <a class="btn btn-primary" href="{{ route('roles.index') }}> Back</a>  
        </div>  
    </div>  
</div>  
  
<div class="row">  
    <div class="col-xs-12 col-sm-12 col-md-12">  
        <div class="form-group">  
            <strong>Name:</strong>
```

```
    {{ $role->name }}  
  </div>  
</div>  
<div class="col-xs-12 col-sm-12 col-md-12">  
  <div class="form-group">  
    <strong>Permissions:</strong>  
    @if(!empty($rolePermissions))  
      @foreach($rolePermissions as $v)  
        <label class="label label-success">{{ $v->name }},</label>  
      @endforeach  
    @endif  
  </div>  
</div>  
</div>  
@endsection
```

Shares



## resources/views/products/index.blade.php

```
@extends('layouts.app')  
  
@section('content')  
  <div class="row">  
    <div class="col-lg-12 margin-tb">  
      <div class="pull-left">  
        <h2>Products</h2>  
      </div>  
      <div class="pull-right">  
        @can('product-create')  
          <a class="btn btn-success" href="{{ route('products.create') }}"> Create New Pr  
        @endcan  
      </div>  
    </div>  
  </div>  
  
  @if ($message = Session::get('success'))  
    <div class="alert alert-success">
```

```
<p>{{ $message }}</p>
</div>
@endif



| No          | Name                  | Details                 | Action                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------|-----------------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {{ ++\$i }} | {{ \$product->name }} | {{ \$product->detail }} | <form action="{{ route('products.destroy', \$product->id) }}" method="POST">     <a class="btn btn-info" href="{{ route('products.show', \$product->id) }}">Show</a>     @can('product-edit')         <a class="btn btn-primary" href="{{ route('products.edit', \$product->id) }}">Edit</a>     @endcan      @csrf     @method('DELETE')     @can('product-delete')         <button type="submit" class="btn btn-danger">Delete</button>     @endcan </form> |



{!! $products->links() !!}

<p class="text-center text-primary"><small>Tutorial by ItSolutionStuff.com</small></p>
```

Shares



```
@endsection
```

## resources/views/products/create.blade.php

Shares



```
@extends('layouts.app')

@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="pull-left">
                <h2>Add New Product</h2>
            </div>
            <div class="pull-right">
                <a class="btn btn-primary" href="{{ route('products.index') }}"> Back </a>
```

```
</div>
</div>
</div>

@if ($errors->any())
    <div class="alert alert-danger">
        <strong>Whoops!</strong> There were some problems with your input.<br><br>
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif

<form action="{{ route('products.store') }}" method="POST">
    @csrf

    <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Name:</strong>
                <input type="text" name="name" class="form-control" placeholder="Name">
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Detail:</strong>
                <textarea class="form-control" style="height:150px" name="detail" placeholder="Detail"></textarea>
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12 text-center">
            <button type="submit" class="btn btn-primary">Submit</button>
        </div>
    </div>
</form>
```

Shares



```
<p class="text-center text-primary"><small>Tutorial by Itsolutionstuff.com</small></p>
@endsection
```

## resources/views/products/edit.blade.php

Shares

```
@extends('layouts.app')

@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="pull-left">
                <h2>Edit Product</h2>
            </div>
            <div class="pull-right">
                <a class="btn btn-primary" href="{{ route('products.index') }}"> Back</a>
            </div>
        </div>
    </div>

@if ($errors->any())
    <div class="alert alert-danger">
        <strong>Whoops!</strong> There were some problems with your input.<br><br>
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif

<form action="{{ route('products.update', $product->id) }}" method="POST">
    @csrf
    @method('PUT')
```



Shares



```
<div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Name:</strong>
            <input type="text" name="name" value="{{ $product->name }}" class="form-control" />
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Detail:</strong>
            <textarea class="form-control" style="height:150px" name="detail" placeholder="Enter product detail..."/>
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12 text-center">
        <button type="submit" class="btn btn-primary">Submit</button>
    </div>
</div>

</form>

<p class="text-center text-primary"><small>Tutorial by ItSolutionStuff.com</small></p>
```

@endsection

## resources/views/products/show.blade.php

```
@extends('layouts.app')

@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="pull-left">
                <h2> Show Product</h2>
            </div>
            <div class="pull-right">
                <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>
            </div>
        </div>
    </div>
```

```
</div>
</div>
</div>

<div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Name:</strong>
            {{ $product->name }}
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Details:</strong>
            {{ $product->detail }}
        </div>
    </div>
</div>
@endsection
<p class="text-center text-primary"><small>Tutorial by Itsolutionstuff.com</small></p>
```

Shares



## Step 10: Create Seeder For Permissions and AdminUser

In this step we will create seeder for permissions, Right now we have fixed permission so we create using seeder as listed bellow, but if you can add more permission as you want:

1.role-list

2.role-create

3.role-edit

4.role-delete

5.product-list

6.product-create

7.product-edit

8.product-delete

Shares



So, first create seeder using bellow command:

```
php artisan make:seeder PermissionTableSeeder
```

And put bellow code in PermissionTableSeeder seeder this way:

### database/seeds/PermissionTableSeeder.php

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Spatie\Permission\Models\Permission;

class PermissionTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $permissions = [
            'role-list',
            'role-create',
            'role-edit',
            'role-delete',
            'product-list',
            'product-create',
            'product-edit',
            'product-delete',
        ];
        foreach ($permissions as $permission) {
            Permission::create(['name' => $permission]);
        }
    }
}
```

```
'role-edit',
'role-delete',
'product-list',
'product-create',
'product-edit',
'product-delete'
];

foreach ($permissions as $permission) {
    Permission::create(['name' => $permission]);
}
}

}
```

Shares



After this we have to run bellow command for run PermissionTableSeeder seeder:

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

Now let's create new seeder for creating admin user.

```
php artisan make:seeder CreateAdminUserSeeder
```

## database/seeds/CreateAdminUserSeeder.php

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use App\Models\User;
use Spatie\Permission\Models\Role;
use Spatie\Permission\Models\Permission;

class CreateAdminUserSeeder extends Seeder
```

```
{  
    /**/  
     * Run the database seeds.  
     *  
     * @return void  
     */  
    public function run()  
    {  
        $user = User::create([  
            'name' => 'Hardik Savani',  
            'email' => 'admin@gmail.com',  
            'password' => bcrypt('123456')  
        ]);  
  
        $role = Role::create(['name' => 'Admin']);  
  
        $permissions = Permission::pluck('id', 'id')->all();  
  
        $role->syncPermissions($permissions);  
  
        $user->assignRole([$role->id]);  
    }  
}
```

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

Now we are ready to run full example of ACL. so let's run our example so run bellow command for quick run:

```
php artisan serve
```

Access By

```
http://localhost:8000/
```

Now you can login with following credential:

Read Also: [Laravel 8 Stripe Payment Gateway Integration Example](#)

Email: admin@gmail.com

Password: 123456

Shares



Now you can run and check.



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



I hope it can help you....



Like 0    0



**Enjoyed it? Help us by sharing...**



Tags : ACL Laravel Laravel 8

← Previous

[Laravel 8 Highcharts Tutorial for Beginner](#)

Next →

[Laravel Livewire File Upload Tutorial](#)



## Hardik Savani

I'm a full-stack developer, entrepreneur and owner of ItSolutionstuff.com. I live in India and I love to write tutorials and tips that can help to other artisan. I am a big fan of PHP, Laravel, Angular, Vue, Node, Javascript, JQuery, Codeigniter and Bootstrap from the early stage. I believe in Hardworking and Consistency.

Shares

\*Hire Me

Follow Me: [X](#) [G](#)

f

X

P

in

✉

### We are Recommending you

- › [Laravel 8 Markdown | Laravel 8 Send Email using Markdown Example](#)
- › [Laravel 8 Multiple Database Connection Example](#)
- › [How to Get Last Executed Query in Laravel 8?](#)
- › [Laravel 8 Autocomplete Search from Database Example](#)
- › [Laravel 8 Inertia JS CRUD with Jetstream & Tailwind CSS](#)
- › [Laravel 8 Yajra Datatables Example Tutorial](#)
- › [Laravel 8 Import Export Excel and CSV File Tutorial](#)
- › [Laravel 8 Livewire CRUD with Jetstream & Tailwind CSS](#)
- › [Laravel 8 Send Mail using Gmail SMTP Server](#)
- › [Laravel 8 PDF | Laravel 8 Generate PDF File using DomPDF](#)

## › Laravel 8 CRUD Application Tutorial for Beginners



Shares

f

X

p

in

✉

Shares



## Popular Posts

- › [Laravel Ajax DELETE Request Example Tutorial](#)
- › [Laravel Unique Validation With Soft Delete Example](#)
- › [Laravel 10 Socialite Login with Twitter Account Example](#)
- › [Laravel Collection Filter Method Example](#)

› Laravel Convert PDF to Image Example

› Laravel 10 Store JSON in Database Example

› Laravel 7 Multiple Image Upload Example

› Laravel 10 Ajax Form Validation Example Tutorial

› Laravel Eloquent Always Load Model Relation Example

› Laravel Jetstream Login with Username or Email Example

› Laravel 8 Guzzle Http Client Request Example

› Laravel 9 ChartJS Chart Example Tutorial

› Laravel React JS CRUD Application Tutorial

› Laravel 8 Factory Tinker Example Tutorial

› Laravel Mailchimp API Integration Example

Shares



## Categories

Laravel

Angular

PHP

Python

jQuery

Ubuntu

Bootstrap

MySql

Javascript

Guest Post

Node JS

Ajax

Codeigniter

Vue.JS

React JS

HTML

Git

Server

JSON

Installation

CSS

.htaccess

Axios

Google API

Typeahead JS

Google Map

Socket.io

SQL

Highcharts

Facebook API

## Latest Posts

- › How to Register Schedule Command in Laravel 11?
- › Laravel 11 New a Health Check Route Example
- › How to Create Custom Middleware in Laravel 11?
- › How to Publish Broadcasting Channels Route File in Laravel 11?

## › How to Create and Use Trait in Laravel 11?

Shares



Shares

f

x

p

in

✉



Subscribe to the Email Newsletter

Shares



Enter Your Email Address....

Subscribe



Follow Laravel News on

f Facebook

X Twitter

in LinkedIn

G Github



# ItsolutionStuff.com

We try to provide best solution for you

---

[Home](#)

[List Of Categories](#)

[List of Tags](#)

[Disclaimer](#)

[Latest Posts](#)

[Contact US](#)

[About US](#)

[Advertisement](#)

---

© 2016-2024 All Rights Reserved • [www.itsolutionstuff.com](http://www.itsolutionstuff.com)