



ItSolutionStuff.com (<https://www.itsolutionstuff.com>)

Laravel 7/6 User Roles and Permissions from Scratch | Laravel 7/6 ACL

By Hardik Savani (<https://twitter.com/HardikSavani19>) | January 25, 2020 | Category : Laravel

Today's leading topic is creating laravel 7/6 roles and permissions tutorial from scratch. we will create laravel 7/6 acl using spatie laravel-permission package. i written step by step tutorial of laravel 7/6 user roles and permissions authentication module.

You just need to follow few step to done laravel 7/6 roles and permissions auth. so let's follow this tutorial step by step.

Spatie role permission composer package provide way to create acl in laravel 6. 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 6 application.

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 (<https://itsolutionstuff.com/post/laravel-52-user-acl-roles-and-permissions-with-middleware-using-entrust-from-scratch-tutorialexample.html>).

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.

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

The screenshot shows a web application interface for managing users. At the top, there is a navigation bar with links for 'Manage Users', 'Manage Role', 'Manage Product', and a dropdown for 'Hardik Savani'. Below the navigation, the title 'Laravel 6 Roles and Permissions from Scratch' is displayed. A green button labeled 'Create New User' is visible. The main content area contains a table with three rows of user data:

No	Name	Email	Roles	Action
1	Karan Patel	karan@gmail.com	User	Show Edit Delete
2	Pares Patel	pares@gmail.com	User	Show Edit Delete
3	Hardik Savani	admin@gmail.com	Admin	Show Edit Delete

A small note at the bottom center says 'Tutorial by ItSolutionStuff.com'.

List Role:

The screenshot shows a web application interface for managing roles. At the top, there is a navigation bar with links for 'Manage Users', 'Manage Role', 'Manage Product', and a dropdown for 'Hardik Savani'. Below the navigation, the title 'Role Management' is displayed. A green button labeled 'Create New Role' is visible. The main content area contains a table with two rows of role data:

No	Name	Action
1	User	Show Edit Delete
2	Admin	Show Edit Delete

A small note at the bottom center says 'Tutorial by ItSolutionStuff.com'.

Create Role:

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:

Create New User

[Back](#)**Name:****Email:****Password:****Confirm Password:****Role:**

Admin
User

[Submit](#)

Tutorial by ItSolutionStuff.com

List Product:

Products

[Create New Product](#)

Product updated successfully

No	Name	Details	Action
1	This is Test	testing app	Show Edit Delete

Tutorial by ItSolutionStuff.com

Step 1: Laravel 6 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
```

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

```
'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
```

Read Also: [Laravel 6 CRUD Application Tutorial \(<https://www.itsolutionstuff.com/post/laravel-6-crud-application-tutorialexample.html>\)](https://www.itsolutionstuff.com/post/laravel-6-crud-application-tutorialexample.html)

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:

```
<?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->bigIncrements('id');
            $table->string('name');
            $table->text('detail');
            $table->timestamps();
        });
    }

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

Now run migration:

```
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/User.php

```
<?php

namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Spatie\Permission\Traits\HasRoles;

class User extends Authenticatable
{
    use Notifiable;
    use 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 = [
        'password', 'remember_token',
    ];

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

app/Product.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

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

Step 5: Add Middleware

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\Middlewares\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 6 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 6 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:

routes/web.php

```
Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');

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

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\User;
use Spatie\Permission\Models\Role;
use DB;
use Hash;

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();
        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);
        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']);
        }else{
            $input = array_except($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');
    }
}
```

app/Http/Controllers/ProductController.php

```
<?php

namespace App\Http\Controllers;

use App\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', ['only' => ['index']]);
        $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([
            '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
     * @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');
    }
}
```



app/Http/Controllers/RoleController.php

```

<?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' => ['index','store']]);
        $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
     * @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);
    $rolePermissions = Permission::join("role_has_permissions","role_has_permissions.permission_id","="
        ->where("role_has_permissions.role_id",$id)
        ->get();

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

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $role = Role::find($id);
    $permission = Permission::get();
    $rolePermissions = DB::table("role_has_permissions")->where("role_has_permissions.role_id",$id)
        ->pluck('role_has_permissions.permission_id','role_has_permissions.permission_id')
        ->all();

    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');
}
```

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

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

Roles Module

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

Product Module

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 6 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" type="text/css">
    <!-- 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 6 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>
                            <li><a class="nav-link" href="{{ route('register') }}>{{ __('Register') }}</a></li>
                        @else
                            <li><a class="nav-link" href="{{ route('users.index') }}>Manage Users</a></li>
                            <li><a class="nav-link" href="{{ route('roles.index') }}>Manage Role</a></li>
                            <li><a class="nav-link" href="{{ route('products.index') }}>Manage Product</a></li>
                            <li class="nav-item dropdown">
                                <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button"
                                    {{ 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>

                                    <form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;">
                                        @csrf
                                    </form>
                                </div>
                            </li>
                        @endguest
                    </ul>
                </div>
            </div>
        </nav>
    </div>
</body>

```

```
@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](#)

```

@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


```

resources/views/users/create.blade.php

```

@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>
</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>
    <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 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>


```

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

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

resources/views/users/edit.blade.php

```

@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 class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Role:</strong>
            {!! Form::select('roles[]', $roles,$userRole, 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() !!}



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


```

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>
</div>
@endsection


```

resources/views/roles/index.blade.php

```

@extends('layouts.app')

@section('content')


<div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2>Role Management</h2>
        </div>
        <div class="pull-right">
            @can('role-create')
                <a class="btn btn-success" href="{{ route('roles.create') }}"> Create New Role</a>
            @endcan
        </div>
    </div>
</div>

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


```

resources/views/roles/create.blade.php

```

@extends('layouts.app')

@section('content')


<div class="col-lg-12 margin-tb">
        <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>

@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' => 'name')) }}<br>
                {{ $value->name }}</label>
            @endforeach
        </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/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>
</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($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, $rolePermissions) ? true : false)}} {{ $value->name }}</label>
            <br/>
            @endforeach
        </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>


```

resources/views/roles/show.blade.php

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

@section('content')


## Show Role



Back



Name:
{{ $role->name }}



Permissions:
@if(!empty($rolePermissions))
@foreach($rolePermissions as $v)
    <label class="label label-success">{{ $v->name }}</label>
@endforeach
@endif


```

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 Product</a>
                @endcan
            </div>
        </div>
    </div>

@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>Details</th>
        <th width="280px">Action</th>
    </tr>
    @foreach ($products as $product)
        <tr>
            <td>{{ ++$i }}</td>
            <td>{{ $product->name }}</td>
            <td>{{ $product->detail }}</td>
            <td>
                <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>
            </td>
        </tr>
    @endforeach
</table>

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

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

```

resources/views/products/create.blade.php

```

@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>

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

```

resources/views/products/edit.blade.php

```

@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')

        <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" required="required"/>
                </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 here" required="required"></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>

    <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 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>

```

Step 10: Handle Exertion Error

Now, in this step we will handle exertion. if you don't have a permission and try to access that page using browser url then you can give message as like bellow:

`add/Exceptions/Handler.php`

```

.....
public function render($request, Exception $exception)
{
    if ($exception instanceof \Spatie\Permission\Exceptions\UnauthorizedException) {
        return response()->json(['User have not permission for this page access.']);
    }

    return parent::render($request, $exception);
}
.....

```

Step 11: 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

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

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]);
        }
    }
}
```

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/PermissionTableSeeder.php

```
<?php

use Illuminate\Database\Seeder;
use App\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: [How to use Laravel 8/7 Authorization using Gates?](#)
[\(https://www.itsolutionstuff.com/post/how-to-use-laravel-6-authorization-using-gatesexample.html\)](https://www.itsolutionstuff.com/post/how-to-use-laravel-6-authorization-using-gatesexample.html)

Email: admin@gmail.com
 Password: 123456

Now you can run and check.

You can also download code from git:

Download Code from Github (<https://github.com/savanihd/Laravel-6-ACL>)

I hope it can help you....

 Tags : ACL (https://www.itsolutionstuff.com/tag/acl.html)	
Example (https://www.itsolutionstuff.com/tag/example.html)	Laravel 6 (https://www.itsolutionstuff.com/tag/laravel-6.html)
Laravel 7 (https://www.itsolutionstuff.com/tag/laravel-7.html)	Tutorial (https://www.itsolutionstuff.com/tag/tutorial.html)



Hardik Savani

My name is 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 artisan. I am a big fan of PHP, Javascript, JQuery, Laravel, Codeigniter, VueJS, AngularJS and Bootstrap from the early stage.

Follow Me: [Github](https://github.com/savanihd) (<https://github.com/savanihd>) [Twitter](https://twitter.com/HardikSavani19) (<https://twitter.com/HardikSavani19>)

***Do you want me hire for your Project Work? Then [Contact US. \(/feedback\)](#)

We are Recommending you:

1. Real Time Event Broadcasting with Laravel 8/7 and Socket.io
[\(<https://www.itsolutionstuff.com/post/real-time-event-broadcasting-with-laravel-6-and-socketioexample.html>\)](https://www.itsolutionstuff.com/post/real-time-event-broadcasting-with-laravel-6-and-socketioexample.html)
2. Send Email with Laravel 7/6 using Markdown Mailable Class
[\(<https://www.itsolutionstuff.com/post/send-email-with-laravel-6-using-markdown-mailable-classexample.html>\)](https://www.itsolutionstuff.com/post/send-email-with-laravel-6-using-markdown-mailable-classexample.html)

3. Laravel 6 Admin Panel Tutorial (<https://www.itsolutionstuff.com/post/laravel-6-admin-panel-tutorialexample.html>)
 4. Laravel 7/6 REST API with Passport Tutorial
(<https://www.itsolutionstuff.com/post/laravel-6-rest-api-with-passport-tutorialexample.html>)
 5. Laravel 7/6 Multi Auth (Authentication) Tutorial
(<https://www.itsolutionstuff.com/post/laravel-6-multi-auth-authentication-tutorialexample.html>)
 6. How to use Yajra Datatables in Laravel 6?
(<https://www.itsolutionstuff.com/post/how-to-use-yajra-datatables-in-laravel-6example.html>)
 7. Laravel 6 CORS Middleware Tutorial
(<https://www.itsolutionstuff.com/post/laravel-6-cors-middleware-tutorialexample.html>)
 8. Laravel 7/6 Ajax CRUD Tutorial (<https://www.itsolutionstuff.com/post/laravel-6-ajax-crud-tutorialexample.html>)
 9. Laravel 6 Authentication Tutorial
(<https://www.itsolutionstuff.com/post/laravel-6-authentication-tutorialexample.html>)
 10. Laravel 6 CRUD Application Tutorial
(<https://www.itsolutionstuff.com/post/laravel-6-crud-application-tutorialexample.html>)
-

Popular Posts

- 👉 Laravel - Barcode generator example using milon/barcode (<https://www.itsolutionstuff.com/post/laravel-5-barcode-generator-example-using-milon-barcode-packageexample.html>)
- 👉 Laravel 7 Yajra Datatables Example (<https://www.itsolutionstuff.com/post/laravel-7-yajra-datatables-exampleexample.html>)

↳ Laravel 5.8 Datatables Tutorial (<https://www.itsolutionstuff.com/post/laravel-58-datatables-tutorialexample.html>)

↳ Laravel 8 Barcode Generator Example (<https://www.itsolutionstuff.com/post/laravel-8-barcode-generator-exampleexample.html>)

↳ How to Store Multiple Select Values in Laravel? (<https://www.itsolutionstuff.com/post/how-to-store-multiple-select-values-in-laravelexample.html>)

↳ Laravel - improve site performance by caching entire response (<https://www.itsolutionstuff.com/post/laravel-improve-site-performance-by-caching-entire-responseexample.html>)

↳ Laravel 6 CRUD Application Tutorial (<https://www.itsolutionstuff.com/post/laravel-6-crud-application-tutorialexample.html>)

↳ Laravel 5 - Full Text Search Example using Scout and Algolia Packages (<https://www.itsolutionstuff.com/post/laravel-5-full-text-search-example-using-scout-and-algolia-packagesexample.html>)

↳ Laravel Eloquent whereBetween() Query Example (<https://www.itsolutionstuff.com/post/laravel-eloquent-wherebetween-query-exampleexample.html>)

↳ How to check table is exists or not in Laravel? (<https://www.itsolutionstuff.com/post/how-to-check-table-is-exists-or-not-in-laravel-5example.html>)

↳ Laravel 8 Livewire CRUD with Jetstream & Tailwind CSS (<https://www.itsolutionstuff.com/post/laravel-8-livewire-crud-with-jetstream-tailwind-cssexample.html>)

Categories

* LARAVEL

([HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/LARAVELEXAMPLE.HTML](https://www.itsolutionstuff.com/category/laravelexample.html))

* ANGULAR

([HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/ANGULAREXAMPLE.HTML](https://www.itsolutionstuff.com/category/angularexample.html))

* BOOTSTRAP ([HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/BOOTSTRAP](https://www.itsolutionstuff.com/category/bootstrap))

* JAVASCRIPT

([HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/JAVASCRIPTEXAMPLE.HTML](https://www.itsolutionstuff.com/category/javascriptexample.html))

* MYSQL ([HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/MYSQLEXAMPLE.HTML](https://www.itsolutionstuff.com/category/mysqlexample.html))

* AJAX ([HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/AJAXEXAMPLE.HTML](https://www.itsolutionstuff.com/category/ajaxexample.html))

* HTML ([HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/HTML](https://www.itsolutionstuff.com/category/html))

* CODEIGNITER	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/CODEIGNITEREXAMPLE.HTML)
REACTJS	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/REACTEXAMPLE.HTML)
JSEXAMPLE.HTML	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/JSEXAMPLE.HTML)
JSON	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/JSONEXAMPLE.HTML)
* GUEST POST	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/GUEST-POSTEXAMPLE.HTML)
INSTALLATION	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/INSTALLATIONEXAMPLE.HTML)
CSS	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/CSSEXAMPLE.HTML)
* GIT	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/GITEXAMPLE.HTML)
* NODE JS	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/NODE-JSEXAMPLE.HTML)
GOOGLE MAP	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/GOOGLE-MAPEXAMPLE.HTML)
MARQUEE	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/MARQUEEEXAMPLE.HTML)
SQLEXAMPLE.HTML	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/SQLEXAMPLE.HTML)
* GOOGLE API	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/GOOGLE-APIEXAMPLE.HTML)
SERVER	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/SERVEREXAMPLE.HTML)
JQUERY UI	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/JQUERYUIEXAMPLE.HTML)
TYPEAHEADJS	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/TYPEAHEAD-JSEXAMPLE.HTML)
SOCKETIO	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/SOCKETIOEXAMPLE.HTML)
AXIOS	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/AXIOSEXAMPLE.HTML)
HIGHCHARTS	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/HIGHCHARTSEXAMPLE.HTML)
ELASTICSEARCH	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/ELASTICSEARCHEXAMPLE.HTML)
WAMPSERVER	(HTTPS://WWW.ITSOLUTIONSTUFF.COM/CATEGORY/WAMPSERVEREXAMPLE.HTML)

Latest Posts

- ↳ Laravel Send SMS to Mobile with Nexmo Example (<https://www.itsolutionstuff.com/post/laravel-send-sms-to-mobile-with-nexmo-exampleexample.html>)
- ↳ MYSQL Query for Data between Two Dates Example (<https://www.itsolutionstuff.com/post/mysql-query-for-data-between-two-dates-exampleexample.html>)
- ↳ Angular 11 Google OAuth Social Login Example (<https://www.itsolutionstuff.com/post/angular-11-google-oauth-social-login-exampleexample.html>)
- ↳ Angular Pipe Under Condition Example (<https://www.itsolutionstuff.com/post/angular-pipe-under-condition-exampleexample.html>)

Angular Pipe for Boolean Value with Yes No Text (<https://www.itsolutionstuff.com/post/angular-pipe-for-boolean-value-with-yes-no-textexample.html>)

Subscribe Your Email address:

Subscribe

Top Links:

- [Home](https://www.itsolutionstuff.com) (<https://www.itsolutionstuff.com>)
- [List Of Categories](https://www.itsolutionstuff.com/categories) (<https://www.itsolutionstuff.com/categories>)
- [Demo Posts](https://www.itsolutionstuff.com/demopost) (<https://www.itsolutionstuff.com/demopost>)
- [Disclaimer](https://www.itsolutionstuff.com/disclaimer) (<https://www.itsolutionstuff.com/disclaimer>)
- [Latest Posts](https://www.itsolutionstuff.com/latestpost) (<https://www.itsolutionstuff.com/latestpost>)
- [Contact US](https://www.itsolutionstuff.com/feedback) (<https://www.itsolutionstuff.com/feedback>)
- [About US](https://www.itsolutionstuff.com/aboutus) (<https://www.itsolutionstuff.com/aboutus>)

© 2016 All Rights Reserved • www.itsolutionstuff.com