

<https://laravelarticle.com>

## Laravel 8 CRUD example - Step by step

Published Sep 09, 2020

**Topic:** CRUD (<https://laravelarticle.com/category/post-on-crud>)

**Author:** Harun

If you are new in Laravel 8 and looking for a step by step tutorial for Laravel 8 CRUD example app then this post will help you to learn how to make a complete CRUD application using Laravel 8. Before starting we have to check our system requirement is okay to use Laravel 8. The Laravel 8 minimum requirements are listed below so that you can confirm either your system is okay or not to install Laravel 8 project for making a Laravel 8 CRUD application.

### Laravel 8 requirements

- PHP >= 7.3
- BCMath PHP Extension
- Ctype PHP Extension
- Fileinfo PHP Extension
- JSON PHP Extension
- Mbstring PHP Extension
- OpenSSL PHP Extension
- PDO PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension

### Steps for making Laravel 8 CRUD

- Step 01: Install Laravel 8
- Step 02: Database Configuration
- Step 03: Make model & migration
- Step 04: Make controller
- Step 05: Define routes

- Step 06: Make views

## Step 01: Install Laravel 8

First, install a fresh new Laravel 8 project. To do that, open your command prompt and run the artisan command below. This command will install and create a new Laravel 8 project for you. Before running this command make sure you have a stable internet connection. This command will take some times depends on your internet connection speed.

```
composer create-project laravel/laravel laravel8-project 8.0
```

[Copy](#)

N.B: Replace the `laravel8-project` with your project name. According to this name, a folder will create in your project directory.

## Step 02: Database Configuration

Now create a database in MySQL via phpMyAdmin or other MySQL clients which you. Now open `.env` file from Laravel 8 project and update the database details.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel8
DB_USERNAME=root
DB_PASSWORD=
```

[Copy](#)

Here `laravel8` is our database name. If your database name different then update it and save. Our project creation finished and the database is ready to use.

## Step 03: Make model & migration

We will make a contact list Laravel 8 CRUD example application. So that we need a contacts table in our database. Here we do not create the table manually. Here we use Laravel migration. When we'll run our migration that will make the table for us. Run the command in your terminal.

```
php artisan make:model Contact -m
```

[Copy](#)

This command will make a `Contact.php` model class file into the `app/Models` directory of our Laravel 8 project and a migration file will be created into the database migrations directory.

Now open the migration file from `database/migrations` directory of your Laravel 8 project and replace the code with below.

```
<?php

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

class CreateContactsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('contacts', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->string('name');
            $table->string('email');
            $table->string('phone');
            $table->timestamps();
        });
    }

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

Copy

Our migration file is ready. Now run the migration with this command. This command will create our tables in our database.

```
php artisan migrate
```

Copy

## Step 04: Make controller

In our controller, all our business login will be coded to make Laravel 8 CRUD system. To make the controller run the command.

```
php artisan make:controller ContactController
```

Copy

By this command, a file will be created in `app/Http/Controllers` name with `ContactController.php`. Write the code below in the `ContactController.php`

```
<?php namespace App\Http\Controllers;

use App\Contact;
use Response;
use App\Http\Requests;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;

class ContactController extends Controller
{

    public function index()
    {
        $data = Contact::orderBy('id','desc')->paginate(10)->setPath('contacts');
        return view('admin.contacts.index',compact(['data']));
    }

    public function create()
    {
        return view('admin.contacts.create');
    }

    public function store(Request $request)
    {
        $request->validate([
            'name' => 'required',
            'email' => 'required|email',
            'phone' => 'required'
        ]);

        Contact::create($request->all());
        return redirect()->back()->with('success','Create Successfully');
    }

    public function show($id)
    {
        $data = Contact::find($id);
        return view('admin.contacts.show',compact(['data']));
    }

    public function edit($id)
    {
        $data = Contact::find($id);
        return view('admin.contacts.edit',compact(['data']));
    }

    public function update(Request $request, $id)
    {
        $request->validate([
            'name' => 'required',
            'email' => 'required|email',
            'phone' => 'required'
        ]);

        Contact::where('id',$id)->update($request->all());
        return redirect()->back()->with('success','Update Successfully');
```

```
}

public function destroy($id)
{
    Contact::where('id',$id)->delete();
    return redirect()->back()->with('success','Delete Successfully');
}

}
```

## Step 05: Define routes

Open the `web.php` file from routes folder and write the routes like below.

```
Route::resource('contacts', 'App\Http\Controllers>ContactController');
```

[Copy](#)

or you can also define route like these

```
use App\Http\Controllers>ContactController;
Route::resource('contacts', ContactController::class);
```

[Copy](#)

Here we are using the Laravel resource route which will make all our required routes that are needed for Laravel 8 CURD example app.

## Step 06: Make views

Here is the final part, We need some forms and HTML markup to show our records and data insert, update. Let's make those views. Create a folder inside views folder name with `contacts` so that all views are related to contact CRUD will be in the same folder and organized.

We need the [Laravel H package \(https://github.com/haruncpi/laravel-h\)](https://github.com/haruncpi/laravel-h) for making HTML form easily. Install it by the composer.

```
composer require haruncpi/laravel-h
```

[Copy](#)

Create an `index.blade.php` to show all our records from the database.

```
@extends('layout')
@section('content')
<div class="col-md-12">

    <div class="table-responsive">
        <table class="table table-bordered table-condensed table-striped">
            <thead>

                <th>ID</th>
                <th>NAME</th>
                <th>EMAIL</th>
                <th>PHONE</th>
                <th>ACTION</th>
            </thead>

            <tbody>
                @foreach($data as $row)
                    <tr>

                        <td>{{ $row->id }}</td>
                        <td>{{ $row->name }}</td>
                        <td>{{ $row->email }}</td>
                        <td>{{ $row->phone }}</td>

                        <td>
                            <a href="{{ route('contacts.edit', $row->id) }}" class="btn btn-primary">
                                <form action="{{ route('contacts.destroy', $row->id) }}" method="post">
                                    @csrf @method('DELETE')
                                    <button class="btn btn-danger" type="submit">Delete</button>
                                </form>
                            </td>
                        </tr>
                    @endforeach
                </tbody>

            </table>
        </div>
        <div>
            <?php echo $data->render(); ?>
        </div>
    </div>

@endsection
```

Create a `create.blade.php` file for insert data.

```
@extends('layout')

@section('content')
{!! F::open(['action' => 'ContactController@store', 'method' => 'POST']) !!}

    <div class="col-md-6">

        <div class="form-group required">
            {!! F::label("NAME") !!}
            {!! F::text("name", null ,["class"=>"form-control","required"=>true]) !!}
        </div>

        <div class="form-group required">
            {!! F::label("EMAIL") !!}
            {!! F::text("email", null ,["class"=>"form-control","required"=>true]) !!}
        </div>

        <div class="form-group required">
            {!! F::label("PHONE") !!}
            {!! F::text("phone", null ,["class"=>"form-control","required"=>true]) !!}
        </div>

        <div class="well well-sm clearfix">
            <button class="btn btn-success pull-right" title="Save" type="submit">Create</button>
        </div>
    </div>

{!! Form::close() !!}
@endsection
```

Create an `edit.blade.php` file to edit data.

```

@extends('layout')

@section('content')
    {!! Form::open(['action' => ['ContactController@update', $data->id], 'method' => 'PUT']) !!}

    <div class="col-md-6">

        <div class="form-group required">
            {!! Form::label("NAME") !!}
            {!! Form::text("name", $data->name , ["class" => "form-control", "required"]) !!}
        </div>

        <div class="form-group required">
            {!! Form::label("EMAIL") !!}
            {!! Form::text("email", $data->email , ["class" => "form-control", "required"]) !!}
        </div>

        <div class="form-group required">
            {!! Form::label("PHONE") !!}
            {!! Form::text("phone", $data->phone , ["class" => "form-control", "required"]) !!}
        </div>

        <div class="well well-sm clearfix">
            <button class="btn btn-success pull-right" title="Save" type="submit">Update</button>
        </div>
    </div>

    {!! Form::close() !!}
@endsection

```

Copy

Now our Laravel 8 CRUD app is ready to use. To test the Laravel 8 CRUD app operation first, run the server by `php artisan serve` command and then open your browser and browse <http://localhost:8000/contacts> (<http://localhost:8000/contacts>)

Hope this step by step tutorial on Laravel 8 CRUD app will help you to make your won CRUD system using Laravel 8. If you find this tutorial helpful then please share this with others.

**Share on**

Facebook

(<https://www.facebook.com/sharer/sharer.php?u=https://laravelarticle.com/laravel-8-crud-example>)

Twitter

([https://www.twitter.com/intent/tweet?text=Laravel 8 CRUD example - Step by step&url=https://laravelarticle.com/laravel-8-crud-example](https://www.twitter.com/intent/tweet?text=Laravel+8+CRUD+example+-+Step+by+step&url=https://laravelarticle.com/laravel-8-crud-example))

LinkedIn

(<https://www.linkedin.com/sharing/share-offsite/?url=https://laravelarticle.com/laravel-8-crud-example>)



# Newsletter

To get new article, Please subscribe with your e-mail

Email address	Submit
---------------	--------

## Topics

Articles (<https://laravelarticle.com/category/post-on-articles>)


Packages (<https://laravelarticle.com/category/post-on-packages>)

Tools (<https://laravelarticle.com/category/post-on-tools>)

CRUD (<https://laravelarticle.com/category/post-on-crud>)

Laravel SEO (<https://laravelarticle.com/category/post-on-laravel-seo>)

## Related Post - Latest Post (<https://laravelarticle.com>)

 Laravel Barcode generation tutorial

(<https://laravelarticle.com/laravel-barcode-tutorial>)

### Laravel Barcode generation tutorial

(<https://laravelarticle.com/laravel-barcode-tutorial>)

Jun 29, 2020 (<https://laravelarticle.com/laravel-barcode-tutorial>) Articles

(<https://laravelarticle.com/category/post-on-articles>)

 Laravel Inertia Js CRUD tutorial


(<https://laravelarticle.com/laravel-inertia-js-crud-tutorial>)

## Laravel Inertia Js CRUD tutorial

(<https://laravelarticle.com/laravel-inertia-js-crud-tutorial>)

Mar 22, 2020 (<https://laravelarticle.com/laravel-inertia-js-crud-tutorial>) CRUD

(<https://laravelarticle.com/category/post-on-crud>)

 Laravel 7 CRUD tutorial

(<https://laravelarticle.com/laravel-7-crud-tutorial>)

## Laravel 7 CRUD tutorial

(<https://laravelarticle.com/laravel-7-crud-tutorial>)

Mar 01, 2020 (<https://laravelarticle.com/laravel-7-crud-tutorial>) CRUD

(<https://laravelarticle.com/category/post-on-crud>)

 Laravel Livewire CRUD tutorial

(<https://laravelarticle.com/laravel-livewire-crud-tutorial>)

## Laravel Livewire CRUD tutorial

(<https://laravelarticle.com/laravel-livewire-crud-tutorial>)

Feb 12, 2020 (<https://laravelarticle.com/laravel-livewire-crud-tutorial>) CRUD

(<https://laravelarticle.com/category/post-on-crud>)

 Laravel 6 jQuery Ajax CRUD tutorial

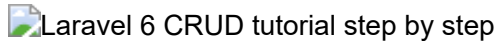
(<https://laravelarticle.com/laravel-jquery-ajax-crud-tutorial>)

## Laravel 6 jQuery Ajax CRUD tutorial

(<https://laravelarticle.com/laravel-jquery-ajax-crud-tutorial>)

Feb 11, 2020 (<https://laravelarticle.com/laravel-jquery-ajax-crud-tutorial>) CRUD

(<https://laravelarticle.com/category/post-on-crud>)



Laravel 6 CRUD tutorial step by step

(<https://laravelarticle.com/laravel-6-crud-tutorial>)

## Laravel 6 CRUD tutorial step by step

(<https://laravelarticle.com/laravel-6-crud-tutorial>)

Nov 05, 2019 (<https://laravelarticle.com/laravel-6-crud-tutorial>) CRUD

(<https://laravelarticle.com/category/post-on-crud>)

---

Home (<https://laravelarticle.com>) | About (<https://laravelarticle.com/page/about>) | Faq  
(<https://laravelarticle.com/page/faq>) | Terms (<https://laravelarticle.com/page/terms>) | Privacy  
Policy (<https://laravelarticle.com/page/privacy-policy>) | Sitemap  
(<https://laravelarticle.com/page/sitemap>) | Contact (<https://laravelarticle.com/page/contact>)

© Copyright 2020 | all rights reserved