



Search



★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



How to install Laravel project you cloned from Git



Sidick A · [Follow](#)

5 min read · Mar 17, 2020



Listen



Share



More



LARAVEL

Install your clone project
from **Git**

Laravel is a powerful php framework and as its slogan says it is ‘The PHP Framework for Web Artisans’.

If you forgot steps or commands during the installation of your Laravel projects, this tutorial is for you.

Here is a guide to install a Laravel project on a Linux or windows server or environment from git.

1- Login with the right permissions (optional)

If you are logged in as root or superadmin on your server, make sure you change your user to the default web user for your project.

Simply do this:

- `Su yourusername`

2- Clone your repo

Whether your project is hosted on [Github](#), [Gitlab](#), [Bitbucket](#) or other, it all comes back to GIT, so we can use git on our local machine or server to clone our project.

- `git clone linktoyourpo.com/project name`

Note that you can clone your project in 2 ways, one in HTTPS mode and one in SSH mode.

PRO TIP: You can clone your project from a specific branch by doing

`git clone -b nameofthebranch linktoyourepo.com/projectname`

3- Access your project directory

After your clone, a new folder with the name of your project should be created in your current location.

You just need to access it by making:

- `cd projectname`

4- Install the project dependencies from compose

Each time you clone a new Laravel project, you have to install all the dependencies of the project. This is what allows you to install Laravel itself, among other packages needed to start your application.

When we run composer, it checks the composer.json file that is in your repo and lists all the composer packages your repo needs. As these packages are constantly changing, the source code is usually not submitted to git, thanks to the .gitignore that should always contain your vendor directory.

So to install all the necessary source code, we run composer with the following command.

- `composer install`

5- Install NPM dependencies (optional)

This is exactly like the previous step with the only difference that it will allow you to install Vue.js, Bootstrap.css, Lodash and Laravel Mix etc...

In short, instead of installing PHP code as in the previous step, it's a matter of installing the required Javascript (or Node) packages.

The list of packages needed in this case are listed in the 'packages.json' file.

If your project doesn't use vue.js, node or other you can skip this step, otherwise you have to do :

- `npm install`

Others prefer Yarn, if that's your case, just do it.

- `Yarn`

6- Create a copy of your .env file

The .env files are generally not submitted to your repo, if this is not the case I invite you to correct this for security reasons.

But there is an example .env file, which is a template of the .env file that every laravel project needs to start.

So we will make a copy of the .env.example file and create an .env file that we can fill in with our configuration settings.

- `cp .env.example .env`

7- Generate your encryption key

Laravel requires that you have an encryption key for each of your applications, this is usually randomly generated and stored in your .env file. The application will use this encryption key to encrypt various elements of your application, such as cookies, password hashes and many other elements.

Fortunately Laravel's command line tools allows you to generate this key very easily. In the terminal, we can execute this command to generate this key.

- `php artisan key:generate`

Close and open your .env file again, you should notice that your key was generated in the variable: APP_KEY

8- Some practical commands (Optional)

If your project does not contain a database at this level it should already be working.

If this is the case — Perfect: everything works like clockwork.



Otherwise try these commands that can help:

Set the right permissions on all directories and files in your project by simply running

- `chmod 755 -R nameofyourproject/`
- `chmod -R o+w nameofyourproject/storage`

Clean up your project

- `php artisan cache:clear`
- `php artisan view:clear`
- `php artisan config:clear`

Sometimes you have to add to your `/public/` or `/index.php/` url

This isn't over, what if you have a database?

Well, let's get on with it.

9- Create an empty database for your project

Create an empty database for your project using any database tools you prefer (My favorite is Datagrip for Mac, but sometimes I use DBForge, or Mysql Workbench or even good old Phpmyadmin).

10- Configure your .env file to allow a connection to the database

We will want to allow Laravel to connect to the database you just created in the previous step. To do this we need to add the connection references in the .env file and Laravel will take care of the connection from there.

In the .env file, fill in the options DB_HOST, DB_PORT, DB_DATABASE, DB_USERNAME and DB_PASSWORD so that they match the credentials of the database you have just created.

11- Add the tables and contents of your database with migrations or in SQL

Migrations in Laravel allow you to have all your DB architecture in your code and with a simple command line you can recreate all your tables.

- `php artisan migrate`

This command creates the structure of your database but does not fill any tables.

If your repository has a seed file, this is the time to run it, in order to fill your database with startup or dummy data.

- `php artisan db:seed`

PRO TIP: You can combine the 2 previous commands into one command which is this one:

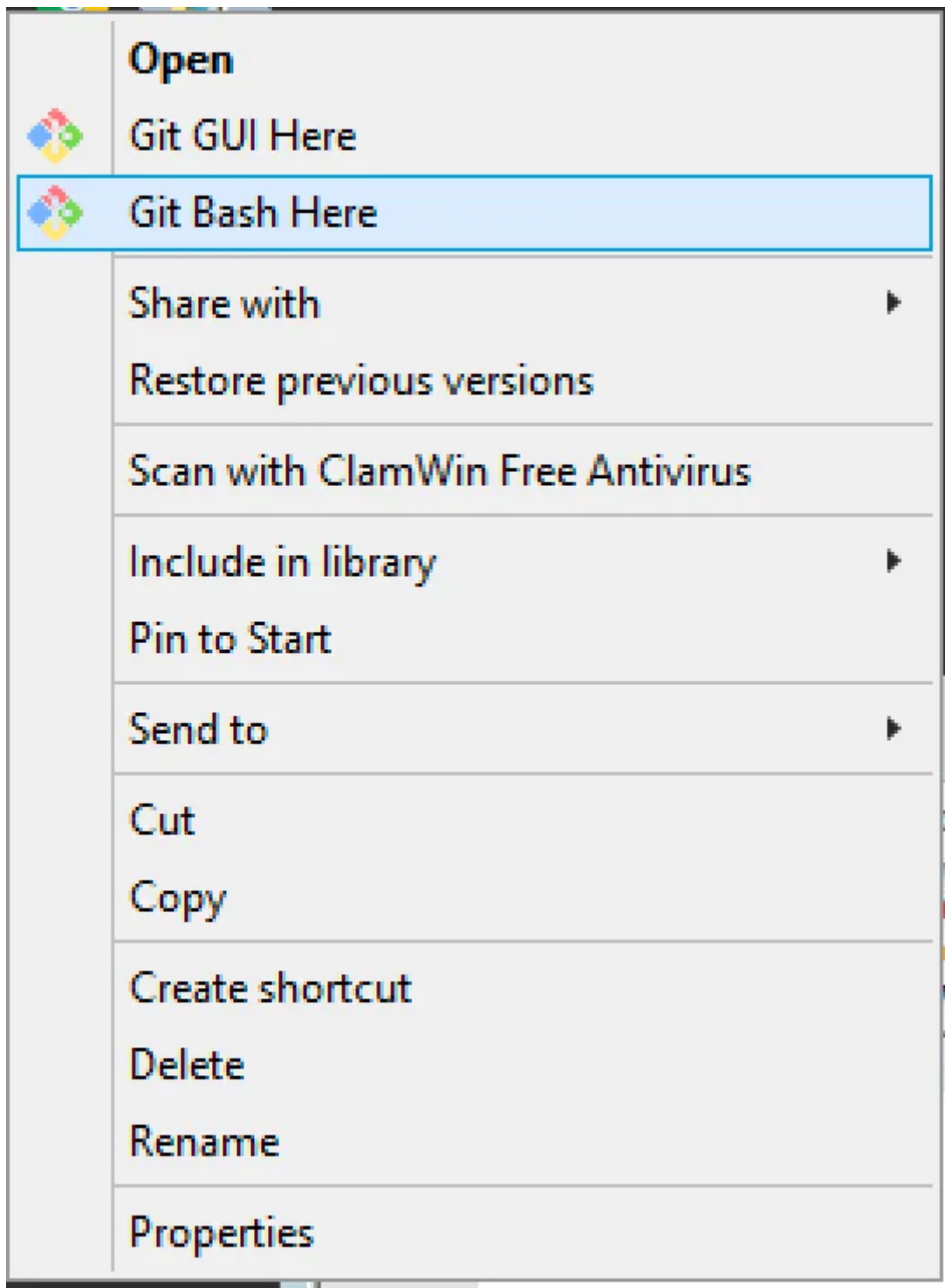
- `php artisan migrate:fresh --seed`

Migrations are super handy, but if you don't have one you can simply import the sql file from your old database with your favorite database tools. This will create all your tables with your contents.

Conclusion

That's all you need to start a project. Of course, some projects have specific steps that apply only to that project, but the steps I've described above are the necessary steps you'll need to follow to start any standard Laravel project from a git clone.

It's essentially the same thing on windows, unless you don't have a batch terminal. This would be surprising because if you install git on a windows server or machine, then right-click in your folder, you should get the 'git bash here' option and you're done.



If you encounter any difficulties, leave a comment and our team will be happy to help you.

I hope this has helped you get started on your next project. Don't forget to subscribe to our newsletter and follow us on social networks for more great tutorials on popular web technologies and marketing techniques.

If you need help with your web security needs contact us [here](#) and our experts will be happy to help you.

Read the original article here: <https://oshara.ca/en/blog/how-to-install-laravel-project-you-cloned-from-git>

Laravel

PHP

Git

Oshara

Montreal



Follow

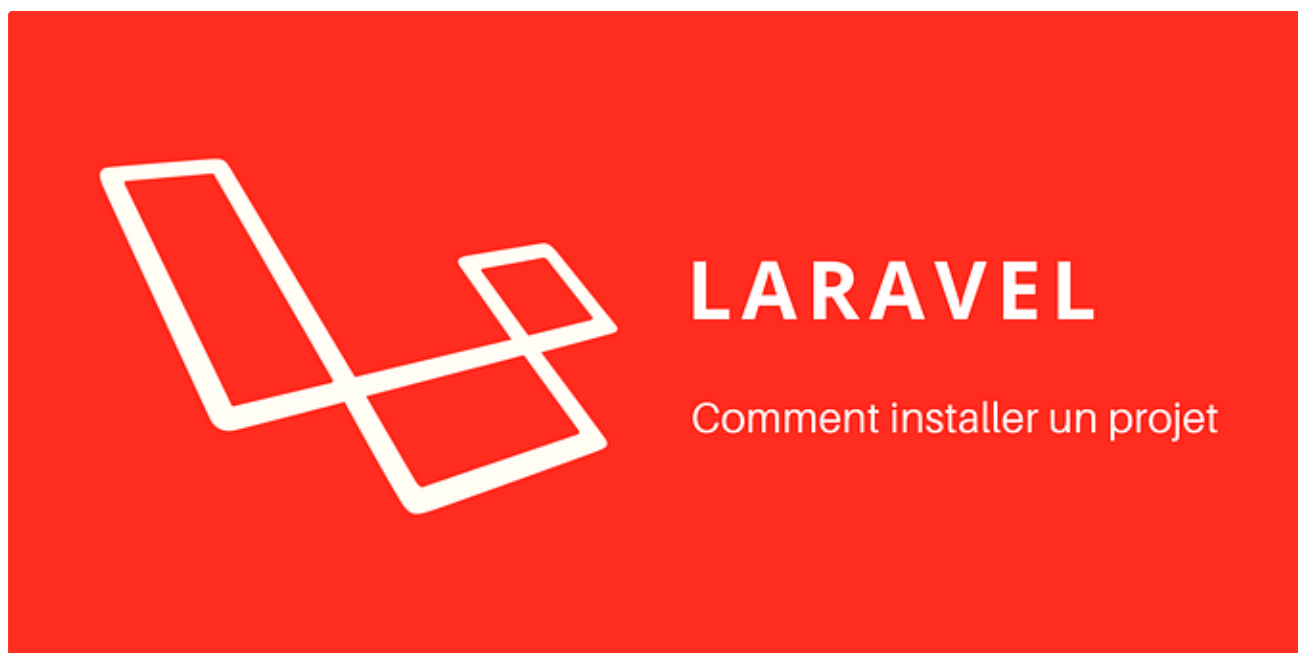


Written by Sidick A

1 Follower

Problem solver, Serial Entrepreneur, Full stack developer I help businesses integrate right technology to solve problems osor.to/sidick

More from Sidick A

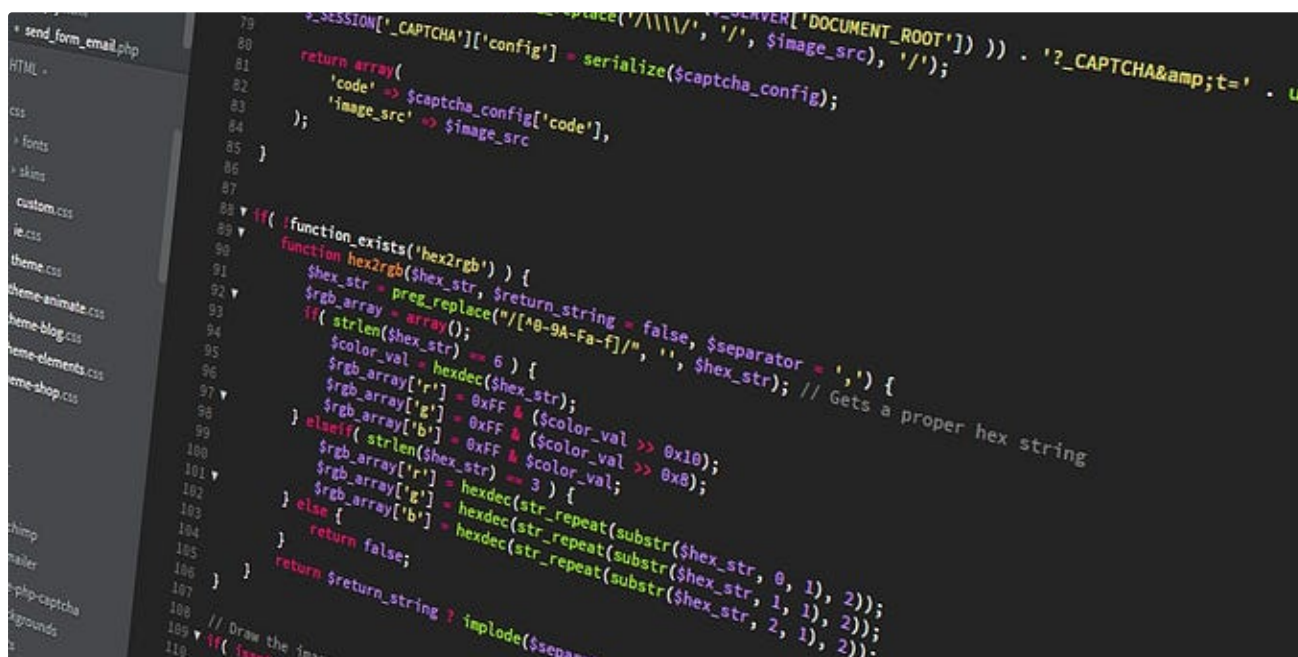
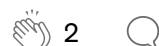


Sidick A

Comment installer un projet Laravel que vous avez cloné depuis Git

Laravel est un Framework php puissant et comme son slogan le mentionne c'est vraiment le Framework PHP pour les artisans du web — 'The PHP...

6 min read · Mar 17, 2020



 Sidick A

How to fix Joomla Error 0 'Cannot open file for writing log while updating'?

Did you already try updating your Joomla website and you got this error :

3 min read · Jan 21, 2020



 Sidick A

Pourquoi avoir un certificat SSL sur tout vos projets web ?

Est-ce nécessaire d'avoir un site web ou une application web en HTTPS ou plus encore souvent appelé un certificat SSL sur son site web ou...

7 min read · Jan 20, 2020





 Sidick A

Quels solutions web choisir pour votre site internet?

Pour lire l'article original, cliquez [ici](#)

7 min read · Jul 7, 2020



See all from Sidick A

Recommended from Medium



Pratik Raman

Building a CRUD Application in Laravel 10: A Step-by-Step Guide

Introduction:

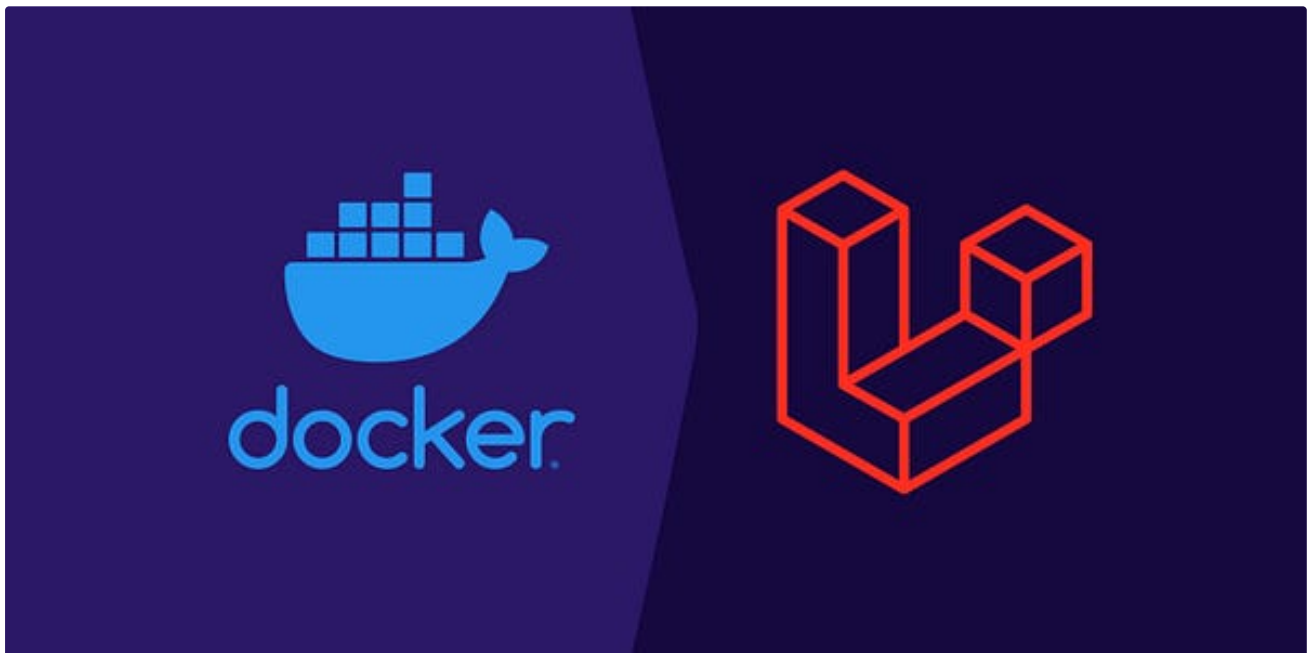
3 min read · Sep 21, 2023




3



1





 Adedayo Adedoyin

Dockerizing a Laravel Project: A Step-by-Step Guide

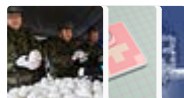
The easiest way to dockerize your Laravel Project

6 min read · Aug 25, 2023

 12 


Lists



Staff Picks

547 stories · 597 saves



 Oliver Samuel

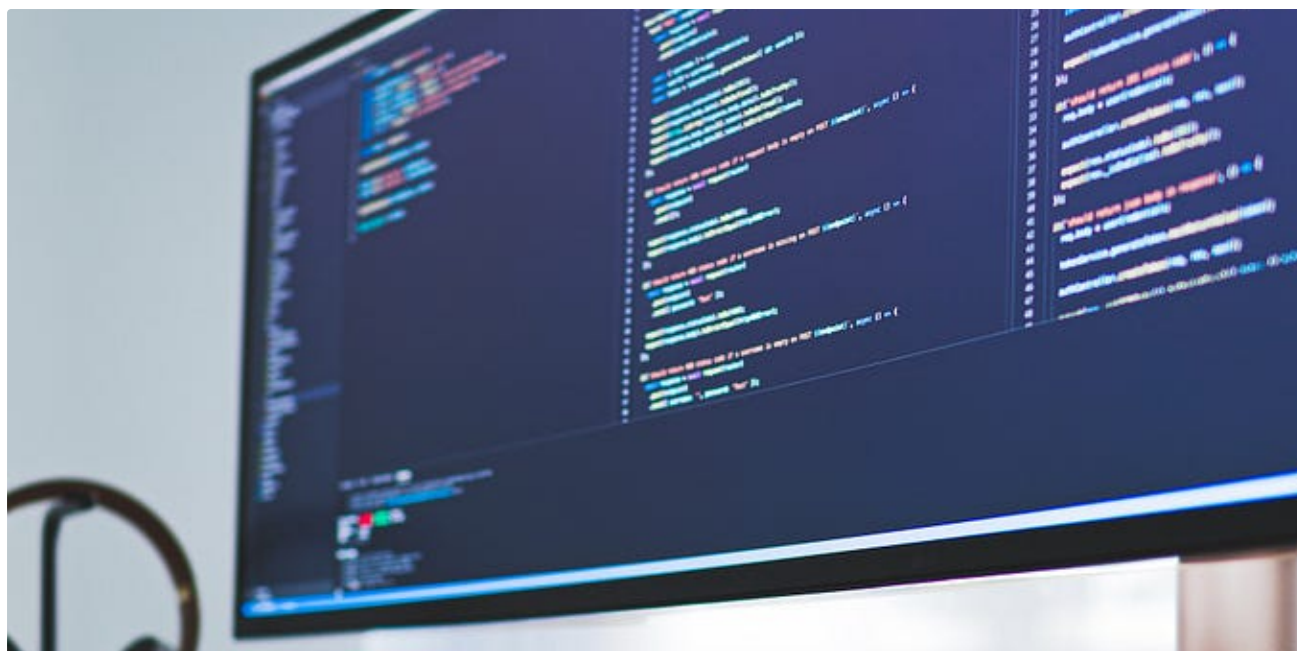
Top 10 Laravel Packages You Should Know in 2024

Laravel, the sleek and expressive PHP framework, continues to enable developers to create sophisticated online apps quickly and...

4 min read · Dec 18, 2023

 248  3



 Vitalii Shloda

Setting Up a Laravel 10 Development Environment with Docker

With the growing trend of containerization in web development, Docker has become an indispensable tool for developers. Docker allows for...

3 min read · Jul 7, 2023




38



1



 Himanshu Pandey

Laravel Testing

Introduction: — In Laravel, tests are directly contain the two directories by default: 1. Feature testing 2. Unit testing

2 min read · Dec 16, 2023



 Martin Tonev 

Writing Code like a Senior Developer in Laravel

Laravel has emerged as one of the prominent PHP frameworks for building elegant applications

4 min read · Dec 22, 2023



231



8



See more recommendations