composer create-project laravel/laravel “name” - - prefer-dist

**notes**

Join Laracon EU in Amsterdam on August 29th - 31st. Tickets are now available! [**Get your ticket today!**](https://laracon.eu/2018)

[Laravel](https://laravel.com/)



* [Documentation](https://laravel.com/docs)
* [Laracasts](https://laracasts.com/)
* [News](https://laravel-news.com/)
* [Partners](https://laravel.com/partners)
* [Forge](https://forge.laravel.com/)
* [Ecosystem](https://laravel.com/docs/4.2/quick)

4.2

[](https://laravel.com/)

* [**Home**](https://laravel.com/)
* [**Laracasts**](https://laracasts.com/)
* [**News**](https://laravel-news.com/)
* [**Partners**](https://laravel.com/partners)
* [**Forge**](https://forge.laravel.com/)
* [**Ecosystem**](https://laravel.com/docs/4.2/quick)

**Documentation**

* **Preface**
  + [Introduction](https://laravel.com/docs/4.2/introduction)
  + [Quickstart](https://laravel.com/docs/4.2/quick)
  + [Release Notes](https://laravel.com/docs/4.2/releases)
  + [Upgrade Guide](https://laravel.com/docs/4.2/upgrade)
  + [Contribution Guide](https://laravel.com/docs/4.2/contributions)
  + [API Documentation](https://laravel.com/api/4.2/)
* **Getting Started**
  + [Installation](https://laravel.com/docs/4.2/installation)
  + [Configuration](https://laravel.com/docs/4.2/configuration)
  + [Homestead](https://laravel.com/docs/4.2/homestead)
  + [Request Lifecycle](https://laravel.com/docs/4.2/lifecycle)
  + [Routing](https://laravel.com/docs/4.2/routing)
  + [Requests & Input](https://laravel.com/docs/4.2/requests)
  + [Views & Responses](https://laravel.com/docs/4.2/responses)
  + [Controllers](https://laravel.com/docs/4.2/controllers)
  + [Errors & Logging](https://laravel.com/docs/4.2/errors)
* **Learning More**
  + [Authentication](https://laravel.com/docs/4.2/security)
  + [Billing](https://laravel.com/docs/4.2/billing)
  + [Cache](https://laravel.com/docs/4.2/cache)
  + [Core Extension](https://laravel.com/docs/4.2/extending)
  + [Events](https://laravel.com/docs/4.2/events)
  + [Facades](https://laravel.com/docs/4.2/facades)
  + [Forms & HTML](https://laravel.com/docs/4.2/html)
  + [Helpers](https://laravel.com/docs/4.2/helpers)
  + [IoC Container](https://laravel.com/docs/4.2/ioc)
  + [Localization](https://laravel.com/docs/4.2/localization)
  + [Mail](https://laravel.com/docs/4.2/mail)
  + [Package Development](https://laravel.com/docs/4.2/packages)
  + [Pagination](https://laravel.com/docs/4.2/pagination)
  + [Queues](https://laravel.com/docs/4.2/queues)
  + [Security](https://laravel.com/docs/4.2/security)
  + [Session](https://laravel.com/docs/4.2/session)
  + [SSH](https://laravel.com/docs/4.2/ssh)
  + [Templates](https://laravel.com/docs/4.2/templates)
  + [Unit Testing](https://laravel.com/docs/4.2/testing)
  + [Validation](https://laravel.com/docs/4.2/validation)
* **Database**
  + [Basic Usage](https://laravel.com/docs/4.2/database)
  + [Query Builder](https://laravel.com/docs/4.2/queries)
  + [Eloquent ORM](https://laravel.com/docs/4.2/eloquent)
  + [Schema Builder](https://laravel.com/docs/4.2/schema)
  + [Migrations & Seeding](https://laravel.com/docs/4.2/migrations)
  + [Redis](https://laravel.com/docs/4.2/redis)
* **Artisan CLI**
  + [Overview](https://laravel.com/docs/4.2/artisan)
  + [Development](https://laravel.com/docs/4.2/commands)

[](https://srv.carbonads.net/ads/click/x/GTND42QICT7DV2JUFTY4YKQMCKYD45QUF6ADEZ3JCW7ITK7MFTBIT2QKC6BICK3LCW7DTK3EHJNCLSIZ?segment=placement:laravelcom;)[See how your visitors are really using your website.](https://srv.carbonads.net/ads/click/x/GTND42QICT7DV2JUFTY4YKQMCKYD45QUF6ADEZ3JCW7ITK7MFTBIT2QKC6BICK3LCW7DTK3EHJNCLSIZ?segment=placement:laravelcom;)[ADS VIA CARBON](http://carbonads.net/?utm_source=laravelcom&utm_medium=ad_via_link&utm_campaign=in_unit&utm_term=carbon)

[Expand All](https://laravel.com/docs/4.2/quick)

* Preface
  + [Introduction](https://laravel.com/docs/4.2/introduction)
  + [**Quickstart**](https://laravel.com/docs/4.2/quick)
  + [Release Notes](https://laravel.com/docs/4.2/releases)
  + [Upgrade Guide](https://laravel.com/docs/4.2/upgrade)
  + [Contribution Guide](https://laravel.com/docs/4.2/contributions)
  + [API Documentation](https://laravel.com/api/4.2/)
* Getting Started
* Learning More
* Database
* Artisan CLI

Laravel Quickstart

* [**Installation**](https://laravel.com/docs/4.2/quick#installation)
* [**Local Development Environment**](https://laravel.com/docs/4.2/quick#local-development-environment)
* [**Routing**](https://laravel.com/docs/4.2/quick#routing)
* [**Creating A View**](https://laravel.com/docs/4.2/quick#creating-a-view)
* [**Creating A Migration**](https://laravel.com/docs/4.2/quick#creating-a-migration)
* [**Eloquent ORM**](https://laravel.com/docs/4.2/quick#eloquent-orm)
* [**Displaying Data**](https://laravel.com/docs/4.2/quick#displaying-data)
* [**Deploying Your Application**](https://laravel.com/docs/4.2/quick#deploying-your-application)

[Installation](https://laravel.com/docs/4.2/quick" \l "installation)

Via Laravel Installer

First, download the Laravel installer using Composer.

composer global require "laravel/installer=~1.1"

Make sure to place the ~/.composer/vendor/bin directory in your PATH (or C:\%HOMEPATH%\AppData\Roaming\Composer\vendor\bin if working with Windows) so the laravelexecutable is found when you run the laravel command in your terminal.

Once installed, the simple laravel new command will create a fresh Laravel installation in the directory you specify. For instance, laravel new blog would create a directory named blogcontaining a fresh Laravel installation with all dependencies installed. This method of installation is much faster than installing via Composer.

Via Composer

The Laravel framework utilizes [Composer](http://getcomposer.org/) for installation and dependency management. If you haven't already, start by [installing Composer](http://getcomposer.org/doc/00-intro.md).

Now you can install Laravel by issuing the following command from your terminal:

composer create-project laravel/laravel your-project-name 4.2.\*

This command will download and install a fresh copy of Laravel in a new your-project-namefolder within your current directory.

If you prefer, you can alternatively download a copy of the [Laravel repository from GitHub](https://github.com/laravel/docs/archive/4.2.zip)manually. Next run the composer install command in the root of your manually created project directory. This command will download and install the framework's dependencies.

Permissions

After installing Laravel, you may need to grant the web server write permissions to the app/storage directories. See the [Installation](https://laravel.com/docs/4.2/installation) documentation for more details on configuration.

Serving Laravel

Typically, you may use a web server such as Apache or Nginx to serve your Laravel applications. If you are on PHP 5.4+ and would like to use PHP's built-in development server, you may use the serve Artisan command:

php artisan serve

By default the HTTP-server will listen to port 8000. However if that port is already in use or you wish to serve multiple applications this way, you might want to specify what port to use. Just add the --port argument:

php artisan serve --port=8080

Directory Structure

After installing the framework, take a glance around the project to familiarize yourself with the directory structure. The app directory contains folders such as views, controllers, and models. Most of your application's code will reside somewhere in this directory. You may also wish to explore the app/config directory and the configuration options that are available to you.

[Local Development Environment](https://laravel.com/docs/4.2/quick" \l "local-development-environment)

In the past, configuring a local PHP development environment on your machine was a headache. Installing the proper version of PHP, required extensions, and other needed components is time consuming and confusing. Instead, consider using [Laravel Homestead](https://laravel.com/docs/4.2/homestead). Homestead is a simple virtual machine designed for Laravel and [Vagrant](http://vagrantup.com/). Since the Homestead Vagrant box is pre-packaged with all of the software you need to build robust PHP applications, you can create a virtualized, isolated development environment in seconds. Here is a list of some of the goodies included with Homestead:

* Nginx
* PHP 5.6
* MySQL
* Redis
* Memcached
* Beanstalk

Don't worry, even though "virtualized" sounds complicated, it's painless. VirtualBox and Vagrant, which are Homestead's two dependencies, both include simple, graphical installers for all popular operating systems. Check out the [Homestead documentation](https://laravel.com/docs/4.2/homestead) to get started.

[Routing](https://laravel.com/docs/4.2/quick" \l "routing)

To get started, let's create our first route. In Laravel, the simplest route is a route to a Closure. Pop open the app/routes.php file and add the following route to the bottom of the file:

Route::get('users', function()

{

return 'Users!';

});

Now, if you hit the /users route in your web browser, you should see Users! displayed as the response. Great! You've just created your first route.

Routes can also be attached to controller classes. For example:

Route::get('users', 'UserController@getIndex');

This route informs the framework that requests to the /users route should call the getIndexmethod on the UserController class. For more information on controller routing, check out the [controller documentation](https://laravel.com/docs/4.2/controllers).

[Creating A View](https://laravel.com/docs/4.2/quick" \l "creating-a-view)

Next, we'll create a simple view to display our user data. Views live in the app/views directory and contain the HTML of your application. We're going to place two new views in this directory: layout.blade.php and users.blade.php. First, let's create our layout.blade.php file:

<html>

<body>

<h1>Laravel Quickstart</h1>

@yield('content')

</body>

</html>

Next, we'll create our users.blade.php view:

@extends('layout')

@section('content')

Users!

@stop

Some of this syntax probably looks quite strange to you. That's because we're using Laravel's templating system: Blade. Blade is very fast, because it is simply a handful of regular expressions that are run against your templates to compile them to pure PHP. Blade provides powerful functionality like template inheritance, as well as some syntax sugar on typical PHP control structures such as if and for. Check out the [Blade documentation](https://laravel.com/docs/4.2/templates) for more details.

Now that we have our views, let's return it from our /users route. Instead of returning Users!from the route, return the view instead:

Route::get('users', function()

{

return View::make('users');

});

Wonderful! Now you have setup a simple view that extends a layout. Next, let's start working on our database layer.

[Creating A Migration](https://laravel.com/docs/4.2/quick" \l "creating-a-migration)

To create a table to hold our data, we'll use the Laravel migration system. Migrations let you expressively define modifications to your database, and easily share them with the rest of your team.

First, let's configure a database connection. You may configure all of your database connections from the app/config/database.php file. By default, Laravel is configured to use MySQL, and you will need to supply connection credentials within the database configuration file. If you wish, you may change the driver option to sqlite and it will use the SQLite database included in the app/database directory.

Next, to create the migration, we'll use the [Artisan CLI](https://laravel.com/docs/4.2/artisan). From the root of your project, run the following from your terminal:

php artisan migrate:make create\_users\_table

Next, find the generated migration file in the app/database/migrations folder. This file contains a class with two methods: up and down. In the up method, you should make the desired changes to your database tables, and in the down method you simply reverse them.

Let's define a migration that looks like this:

public function up()

{

Schema::create('users', function($table)

{

$table->increments('id');

$table->string('email')->unique();

$table->string('name');

$table->timestamps();

});

}

public function down()

{

Schema::drop('users');

}

Next, we can run our migrations from our terminal using the migrate command. Simply execute this command from the root of your project:

php artisan migrate

If you wish to rollback a migration, you may issue the migrate:rollback command. Now that we have a database table, let's start pulling some data!

[Eloquent ORM](https://laravel.com/docs/4.2/quick" \l "eloquent-orm)

Laravel ships with a superb ORM: Eloquent. If you have used the Ruby on Rails framework, you will find Eloquent familiar, as it follows the ActiveRecord ORM style of database interaction.

First, let's define a model. An Eloquent model can be used to query an associated database table, as well as represent a given row within that table. Don't worry, it will all make sense soon! Models are typically stored in the app/models directory. Let's define a User.php model in that directory like so:

class User extends Eloquent {}

Note that we do not have to tell Eloquent which table to use. Eloquent has a variety of conventions, one of which is to use the plural form of the model name as the model's database table. Convenient!

Using your preferred database administration tool, insert a few rows into your users table, and we'll use Eloquent to retrieve them and pass them to our view.

Now let's modify our /users route to look like this:

Route::get('users', function()

{

$users = User::all();

return View::make('users')->with('users', $users);

});

Let's walk through this route. First, the all method on the User model will retrieve all of the rows in the users table. Next, we're passing these records to the view via the with method. The with method accepts a key and a value, and is used to make a piece of data available to a view.

Awesome. Now we're ready to display the users in our view!

[Displaying Data](https://laravel.com/docs/4.2/quick" \l "displaying-data)

Now that we have made the users available to our view, we can display them like so:

@extends('layout')

@section('content')

@foreach($users as $user)

<p>{{ $user->name }}</p>

@endforeach

@stop

You may be wondering where to find our echo statements. When using Blade, you may echo data by surrounding it with double curly braces. It's a cinch. Now, you should be able to hit the /users route and see the names of your users displayed in the response.

This is just the beginning. In this tutorial, you've seen the very basics of Laravel, but there are so many more exciting things to learn. Keep reading through the documentation and dig deeper into the powerful features available to you in [Eloquent](https://laravel.com/docs/4.2/eloquent) and [Blade](https://laravel.com/docs/4.2/templates). Or, maybe you're more interested in [Queues](https://laravel.com/docs/4.2/queues) and [Unit Testing](https://laravel.com/docs/4.2/testing). Then again, maybe you want to flex your architecture muscles with the [IoC Container](https://laravel.com/docs/4.2/ioc). The choice is yours!

[Deploying Your Application](https://laravel.com/docs/4.2/quick" \l "deploying-your-application)

One of Laravel's goals is to make PHP application development enjoyable from download to deploy, and [Laravel Forge](https://forge.laravel.com/) provides a simple way to deploy your Laravel applications onto blazing fast servers. Forge can configure and provision servers on DigitalOcean, Linode, Rackspace, and Amazon EC2. Like Homestead, all of the latest goodies are included: Nginx, PHP 5.6, MySQL, Postgres, Redis, Memcached, and more. Forge "Quick Deploy" can even deploy your code for you each time you push changes out to GitHub or Bitbucket!

On top of that, Forge can help you configure queue workers, SSL, Cron jobs, sub-domains, and more. For more information, visit the [Forge website](https://forge.laravel.com/).

* [Documentation](https://laravel.com/docs)

* [Laracasts](https://laracasts.com/)

* [News](https://laravel-news.com/)

* [Partners](https://laravel.com/partners)

* [Forge](https://forge.laravel.com/)

* [Ecosystem](https://laravel.com/docs/4.2/quick)

LARAVEL IS A TRADEMARK OF TAYLOR OTWELL. COPYRIGHT © TAYLOR OTWELL.

[DESIGNED BY](http://jackmcdade.com/)