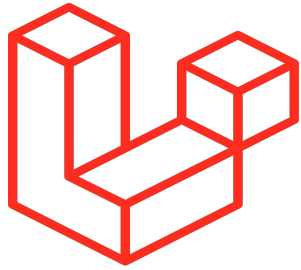


Content

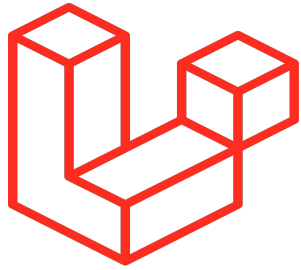
- History
- Laravel Employees
- Composer
- Install Laravel
- Directory Structure
- Routing
- Blade Templating Engine



History

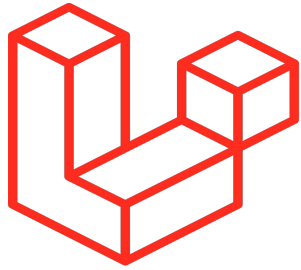
- DHH released first version of ruby on rails on 2004
- Some of php frameworks got inspired by RoR
- In 2010 taylor become dissatisfied with CodeIgniter as

They were slow to catch up new features like namespaces



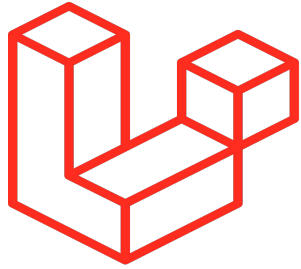
History

- **Laravel 1** (June 2011) included built in support for (authentication - routing - models - views - ..)
- **Laravel 2** (September 2011) included support for (controllers - blade template engine - IOC - ..)
- **Laravel 3** (February 2012) included support for (CLI Artisan - events - migrations - bundles - ..)



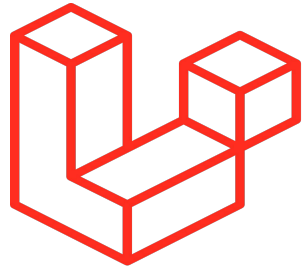
History

- **Laravel 4** (May 2013) taylor rewrote framework from scratch and developed set of components under code name Illuminate , and pulled majority of components from symfony also included support for (seeding - soft delete - ..)
- **Laravel 5** (Feb 2015) new directory tree structure and included support for (socialite - elixir - dotenv -...)



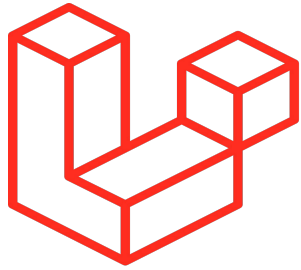
History

- **Laravel 6** (September 2019) introduced [Semantic Versioning](#) and extracted front end scaffolding to [laravel/ui](#) . and introduced [Lazy Collection](#) which allow you to work with very large dataset while keeping memory usage low .
- **Laravel 8** (September 2020) moved to a single release per year [check this blog post for more](#).



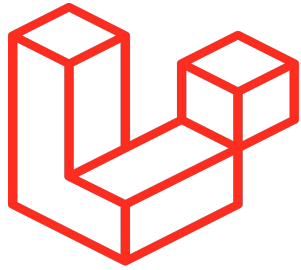
Laravel Employees

- [Taylor Otwel](#) Creator Of Framework
- [Mohamed Said](#) First Laravel Employee
- [Dries Vints](#) Creator Of Popular Open Source [Laravel IO](#)
- [James Brooks](#) Creator Of Popular Open Source [Cachet](#)
- [Nuno Maduro](#) Creator of [Pest PHP Testing Framework](#)



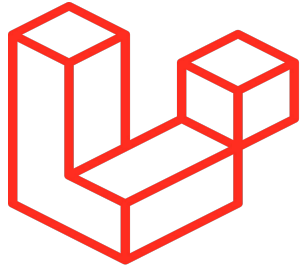
Composer

- Composer is a dependency manager like npm for node or ruby gems for ruby .
- Download From here
<https://getcomposer.org/download/>
- Make composer globally
<https://getcomposer.org/doc/00-intro.md#globally>



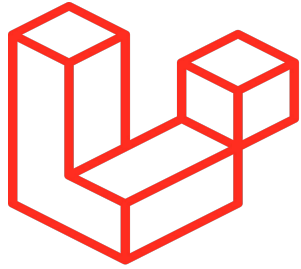
Install Laravel

<https://laravel.com/docs/master/installation#installing-laravel>



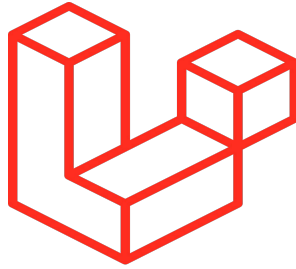
Directory Structure

- Laravel Version ≥ 5.4 you will see :-
- **app/** --> where your application go like (models controllers)
- **bootstrap/** --> files laravel uses to boot every time
- **config/** --> configuration files like (database configuration)
- **database/** --> where (migrations - seeds - factories) exist
- **public/** --> directory where server points to it when serving request it also contains (index.php)
- **resources/** --> contains (views - SaaS - Less files - ..)
- **routes/** --> all routes definitions (Api - console -http)



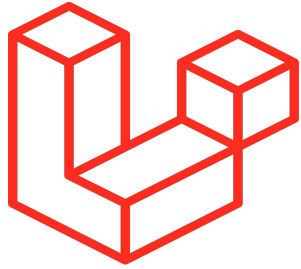
Directory Structure

- `storage/` --> where cache logs compiled files exist
- `tests/` --> where unit & integration test exist
- `vendor/` --> composer packages
- `.env` --> defines environments variables
- `.env.example` --> same as above ,it should be copied when cloning other projects
- `.gitattributes` , `.gitignore` --> git configuration files
- `artisan` --> php script to run artisan commands



Directory Structure

- `composer.json` , `composer.lock` --> contains project dependencies
- `package.json` --> like `composer.json` for frontend assets
- `phpunit.xml` --> configuration for php unit
- `readme.md` --> markdown for laravel introduction
- `server.php` --> it's a server that emulate apache `mod_rewrite`
- `webpack.mix.js` --> used for compiling and mixing frontend assets



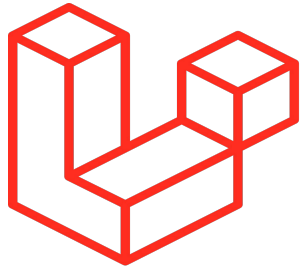
Routing

- Controller's Method

```
Route::get('/posts' , [PostsController::class, 'index']);
```

- Parameters

```
Route::get('/posts/{id}' , [PostsController::class, 'edit']);
```



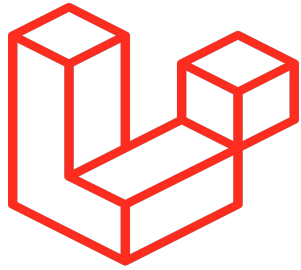
Routing

- Named Routes

```
Route::get('/posts' , [PostsController::class, 'index'])
```

```
->name('posts.index')
```

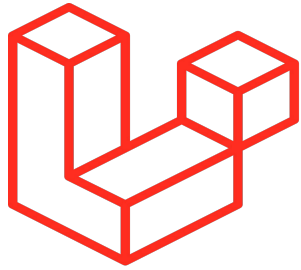
more at <https://laravel.com/docs/master/routing>



Blade

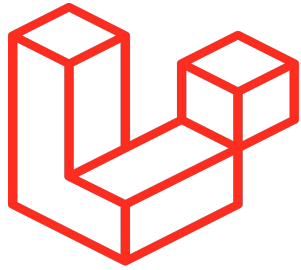
- Blade is inspired by .NET's Razor engine .
- echo data :- {{ \$post }} instead of <?php echo \$post ; ?>
- conditions :-
 @if (\$post->name == 'firstPost')
 // do some stuff
 @endif

<https://laravel.com/docs/master/blade#introduction>



Blade

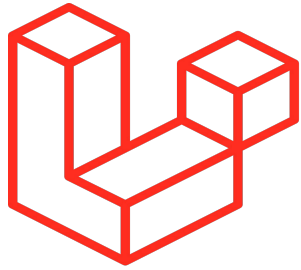
- **looping** :-
`@foreach($posts as $post) //do some stuff @endforeach`
- **inheritance** :- `@extends('layouts.master')`
- **define sections** :-
`@section('content')`
`<h1> hello from the content`
`@endsection`



Blade

- **printing sections** :- `@yield('content')`
- **including** :- `@include('scripts')`
- **Including with parameters** :-
`@include('post.form',['method' => 'POST'])`

more at <https://laravel.com/docs/master/blade#introduction>



Lab 1 (Index & Destroy)

ITI Blog All Posts

Create Post

Please Format
Created At As
Shown in lab 2

Pagination in lab 2

	Title	Posted By	Created At	Actions
1	Learn PHP	Ahmed	2018-04-10	View Edit Delete
2	Solid Principles	Mohamed	2018-04-12	View Edit Delete
3	Design Patterns	Ali	2018-04-13	View Edit Delete

Pagination in lab 2

[Previous](#) [1](#) [2](#) [3](#) [Next](#)

Lab 1 (Create & Store)

ITI Blog All Posts

Title

Description

Post Creator

Create

Lab 1 (Edit & Update)

ITI Blog All Posts

Title

Solid Principles

Description

in this post we are going to know the SOLID Principles , so you should be familiar with OOP

Post Creator

Mohamed

Update

Lab 1 (Show)

Post Info

Title :- Special title treatment

Description :-

With supporting text below as a natural lead-in to additional content.

Post Creator Info

Name :- Ahmed

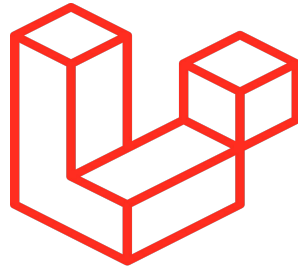
Email :- ahmed@gmail.com

Created At :- Thursday 25th of December 1975 02:15:16 PM

(Search in carbon documentation to know how to format it like this) in lab 2

Lab 1

- We will make **CRUD** operation on **Posts**
- Implement **Resource Controller** methods like screenshots (Index, Create, Store, Edit, Update, Destroy, Show)
<https://laravel.com/docs/master/controllers#resource-controllers>
- **All Posts** is a link that redirect back to **/posts** route
- Use [blade layout](#) to not duplicate navbar across view files
- When submitting a form make sure to redirect back to **/posts** route (For Delete Button Logic Leave It For Lab 2)



Lab 1 (Bonus)

- Install This [Extension For VSCode Or Any Similar Editors](#) & [Put This File In Your Project Root Directory](#) then make the fixer automatically executed on save
- Create a **blade component** called **Button** that accepts type paramter so that it can be called like this

```
<x-button type="primary">View</x-button>  
<x-button type="secondary">Edit</x-button>  
<x-button type="danger">Delete</x-button>
```

[php artisan make:component Button](#)

Then read [this doc link](#) and apply the same concepts on the button component

