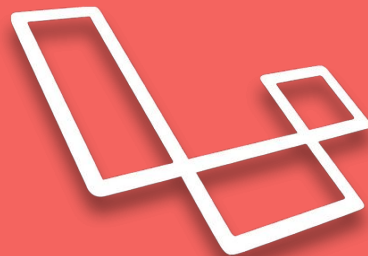


# Content

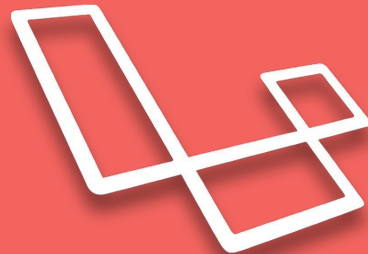
- History
- Composer
- Install Laravel
- Directory Structure
- Routing
- Blade Templating Engine
- Database & Eloquent



# History

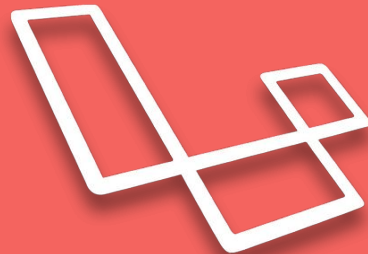
- DHH released first version of ruby on rails on 2004
- Some of php frameworks got inspired by RoR
- In 2010 taylor become dissatisfied with CodeIgniter as

They were slow to catch up new features like namespaces



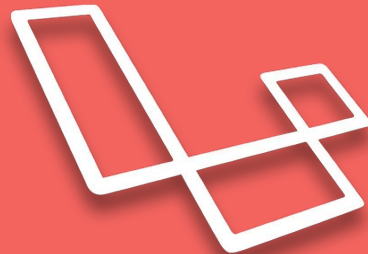
# History

- **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 - ..)



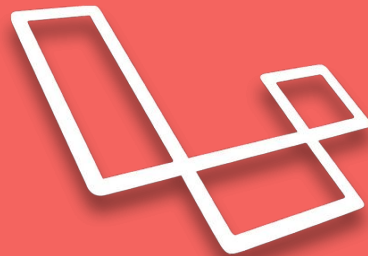
# History

- **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 -...)



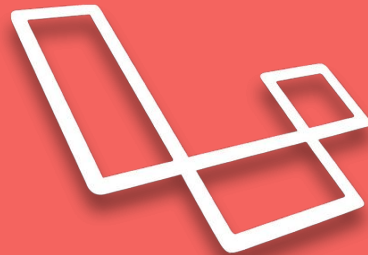
# Composer

- 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>



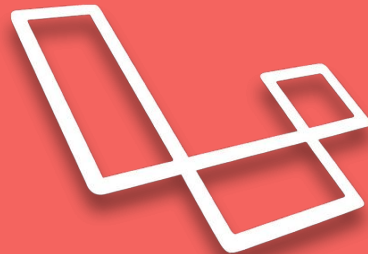
# Install Laravel

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



# Directory Structure

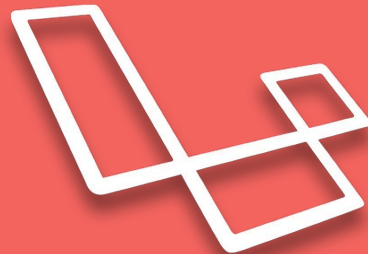
- Laravel Version  $\geq 5.4$  you will see :-
- **app/** --> where your application go like ( models controllers)
- **bootstrap/** --> files laravel uses to boot every time
- **config/** --> configuration files like (database configuration)
- **database/** --> where (migrations - seeds - factories) 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)





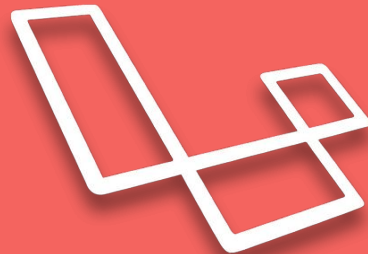
# Directory Structure

- **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



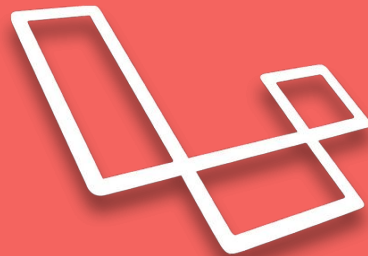
# Routing

- Controller's Method

```
Route::get('/posts' , 'PostsController@index');
```

- Parameters

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

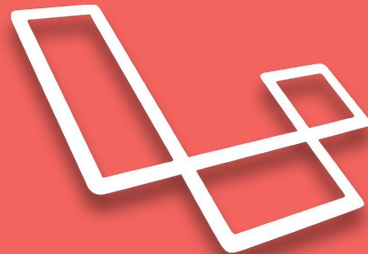


# Routing

- Named Routes

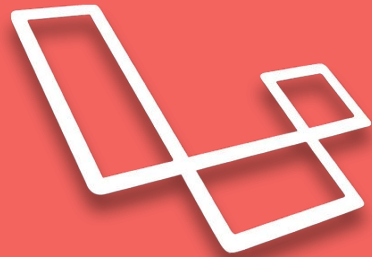
```
Route::get('/posts', [  
    'uses' => 'PostsController@index',  
    'as' => 'posts.index'  
]);
```

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



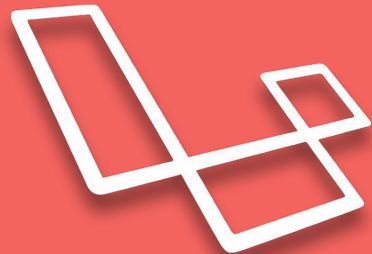
# Blade

- **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



# Blade

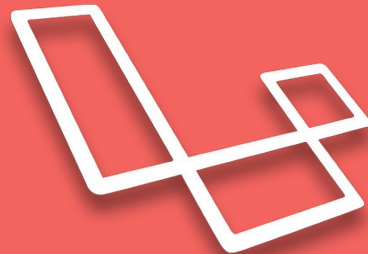
- **looping** :-  
@foreach(\$posts as \$post) //do some stuff @endforeach
- **inheritance** :- @extends('layouts.master')
- **define sections** :-  
@section('content')  
    <h1> hello from the content  
@endsection



# Blade

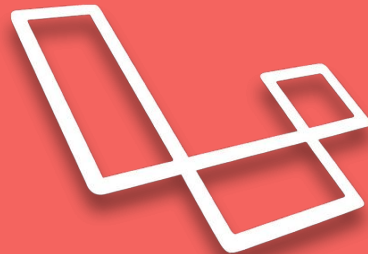
- **printing sections** :- `@yield('content')`
- **including** :- `@include('scripts')`
- **Including with parameters** :-  
`@include('post.form',['method' => 'POST'])`

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



# Database & Eloquent

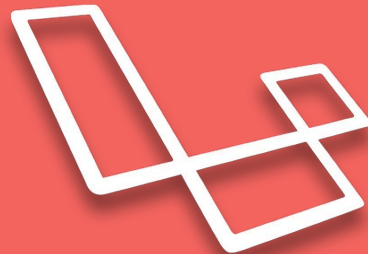
- Laravel ORM is called **Eloquent**
- Database configuration in **.env** file or from **config/database.php**.
- Laravel **migration** helps in making database persistent across multiple machines .
- **DB facade** used to form query builder object





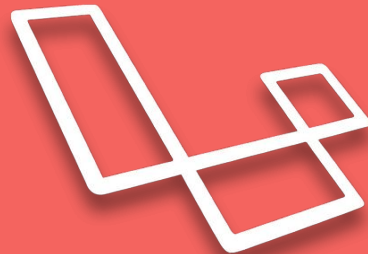
# Database & Eloquent

- `$post = DB::table('posts')->find(20);` //finds a row with id = 20
- `$posts = DB::table('posts')->get();` //get all rows in posts table
- `$singlePost = DB::table('posts')->where('slug', 'FirstPost')->first();` //where conditions to query
- `$firstPost = DB::table('posts')->first();` //gets the first row



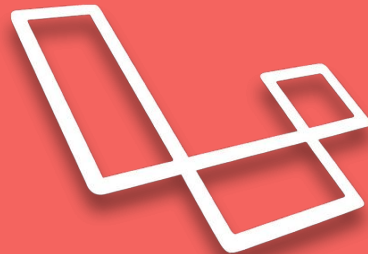
# Database & Eloquent

- `DB::table('posts')->insert([  
    'title' => 'first post title' , 'desc' => 'first post desc']  
);` **//inserting a row**
- `DB::table('posts')->where('id' , 1)->delete();` **//deletes a row**
- `DB::table('posts')->where('id' , 1)  
->update( [ 'title' => 'changed title post' ] );`  
**//update the post title only**



# Database & Eloquent

- `php artisan make:model Post` //create a new model class
- Laravel by default gets the **plural name** of model as a table name and makes query based on that
- `Post::all()` //will search in **posts table** and get all rows
- `Post::create(['title' => 'first post' , 'desc' => 'desc post']);`  
//this will give a **MassAssignmentException** unless you override **\$fillable**



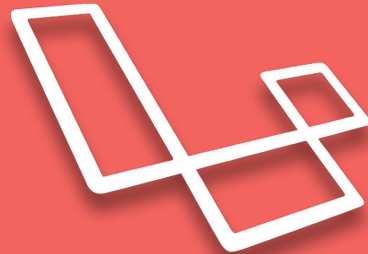
# Database & Eloquent

- `Post::find(25)->update(['title' => 'update post title'])`

*//updates title for post with id 25*

- `Post::where('votes', 23)->delete()`

*//deletes any post have votes with 23*



# Database & Eloquent

- To define a relation in Post model you will define a function in the class for example i want to say post have many sections .

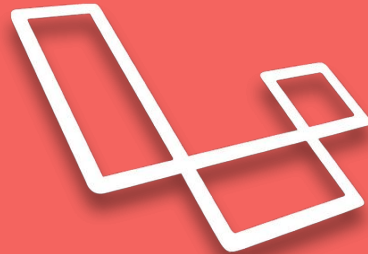
//in Post model class

```
public function sections ()
```

```
{ return $this->hasMany(Section::class);}
```

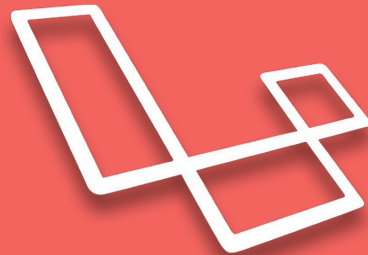
// in controller for example

```
$postSections = Post::find(23)->sections;
```



# Database & Eloquent

- Remember that query results are **collection** objects
- More at Eloquent  
<https://laravel.com/docs/master/eloquent#introduction>
- More at Collections  
<https://laravel.com/docs/master/collections#available-methods>



# Lab 1 (Index & Destroy)

ITI Blog All Posts

Create Post

# Pagination Bonus	Title	Posted By	Created At	Actions
1	Learn PHP	Ahmed	2018-04-10	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
2	Solid Principles	Mohamed	2018-04-12	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
3	Design Patterns	Ali	2018-04-13	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

Pagination Bonus

[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)

ITI Blog All Posts

## Post Info

**Title :-** Special title treatment

**Description :-**

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

## Post Creator Info

**Name :-** Ahmed

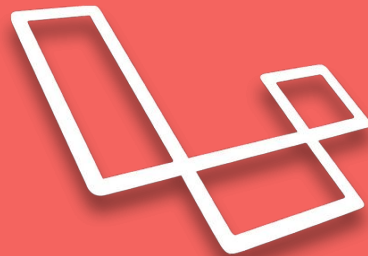
**Email :-** ahmed@gmail.com

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

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

# Lab 1

- We will make **CRUD** operation on **Posts**
- Implement **Resource Controller** methods like screenshots (Index,Create,Edit,Update,Destroy,Show)  
<https://laravel.com/docs/master/controllers#resource-controllers>
- **All Posts** is a link that redirect back to **/posts** route
- When i Click on **Delete** you must show a warning before deleting and i choose between **yes** to confirm Delete or **no**



# Lab 1

- In **Index & Show** page , the **Created At** is formatted , so make sure to read carbon documentation  
<https://carbon.nesbot.com/docs/>
- In **Edit Or Create** Post Creator Field must be drop down list of users
- Bonus :- Add **pagination** to Index page  
<https://laravel.com/docs/master/pagination>
- Bonus :- make **Accessor** that returns human readable carbon to be used in posts/{id}  
<https://laravel.com/docs/master/eloquent-mutators#defining-an-accessor>

