

What is laravel and uses of directories?

Laravel is a free and open-source PHP framework that provides a set of tools and resources to build modern PHP applications. With a complete ecosystem leveraging its built-in features, and a variety of compatible packages and extensions, Laravel has seen its popularity grow rapidly in the past few years, with many developers adopting it as their framework of choice for a streamlined development process.

Laravel provides powerful database tools including an ORM (Object Relational Mapper) called Eloquent, and built-in mechanisms for creating database migrations and seeders. With the command-line tool Artisan, developers can bootstrap new models, controllers, and other application components, which speeds up the overall application development.

- 1.The app directory contains the core code of your application.
almost all of the classes in your application will be in this directory.
- 2.The bootstrap directory contains the app.php file which bootstraps the framework. This directory also houses a cache directory which contains framework generated files for performance optimization such as the route and services cache files.
- 3.The config directory, as the name implies, contains all of your application's configuration files. It's a great idea to read through all of these files and familiarize yourself with all of the options available to you.
- 4.The database directory contains your database migrations,model factories, and seeds. If you wish, you may also use this directory to hold an SQLite database.
5. The lang directory houses all of your application's language files.
- 6.The resources directory contains your views as well as your raw, un-compiled assets such as CSS or JavaScript.
7. The routes directory contains all of the route definitions for your application. By default, several route files are included with Laravel: web.php, api.php, console.php, and channels.php.

The web.php file contains routes that the RouteServiceProvider places in the web middleware group, which provides session state, CSRF protection, and cookie encryption. If your application does not offer a stateless, RESTful API then all your routes will most likely be defined in the web.php file.

The api.php file contains routes that the RouteServiceProvider places in the api middleware group. These routes are intended to be stateless, so requests entering the application through these routes are intended to be authenticated via tokens and will not have access to session state.

The `console.php` file is where you may define all of your closure based console commands. Each closure is bound to a command instance allowing a simple approach to interacting with each command's IO methods. Even though this file does not define HTTP routes, it defines console based entry points (routes) into your application.

The `channels.php` file is where you may register all of the event broadcasting channels that your application supports.

8. The storage directory contains your logs, compiled Blade templates, file based sessions, file caches, and other files generated by the framework. This directory is segregated into `app`, `framework`, and `logs` directories. The `app` directory may be used to store any files generated by your application. The `framework` directory is used to store framework generated files and caches. Finally, the `logs` directory contains your application's log files.

9. The `vendor` directory contains your Composer dependencies.

10. The `app` directory contains a variety of additional directories such as `Console`, `Http`, and `Providers`. Think of the `Console` and `Http` directories as providing an API into the core of your application. The HTTP protocol and CLI are both mechanisms to interact with your application, but do not actually contain application logic. In other words, they are two ways of issuing commands to your application. The `Console` directory contains all of your Artisan commands, while the `Http` directory contains your controllers, middleware, and requests.

10. The `Broadcasting` directory contains all of the broadcast channel classes for your application

11. The `Console` directory contains all of the custom Artisan commands for your application. These commands may be generated using the `make:command` command.

12. The `Http` directory contains your controllers, middleware, and form requests. Almost all of the logic to handle requests entering your application will be placed in this directory.

13. The `Models` directory contains all of your Eloquent model classes. The Eloquent ORM included with Laravel provides a beautiful, simple ActiveRecord implementation for working with your database. Each database table has a corresponding "Model" which is used to interact with that table. Models allow you to query for data in your tables, as well as insert new records into the table.

14. This directory does not exist by default, but will be created for you if you execute the `make:rule` Artisan command. The `Rules` directory contains the custom validation rule objects for your application. Rules are used to encapsulate complicated validation logic in a simple object. For more information, check out

What is usage of artisan in laravel?

Artisan is the name of the command-line interface included with Laravel. It provides a number of helpful commands for your use while developing your application. It is driven by the powerful Symfony Console component.

What is a template engine and Laravel uses what template engine?

Template engines are used when you want to rapidly build web applications that are split into different components. Templates also enable fast rendering of the server-side data that needs to be passed to the application.

For example, you might want to have components such as body, navigation, footer, dashboard, etc.

Laravel uses Blade template engine

What routes do in resources?

Their managing which request do what function for example rendering a view with url changes or or a controller

When declaring a resource route, you may specify a subset of actions the controller should handle instead of the full set of default actions

What use of named routes?

Named routes is an important feature in the Laravel framework. It allows you to refer to the routes when generating URLs or redirects to the specific routes. In short, we can say that the naming route is the way of providing a nickname to the route.

Difference between {{ }} and {!! !!}?

In {{ }} You are not limited to displaying the contents of the variables passed to the view. You may also echo the results of any PHP function. In fact, you can put any PHP code you wish inside of a Blade echo statement

By default, Blade {{ }} statements are automatically sent through PHP's htmlspecialchars function to prevent XSS attacks. If you do not want your data to be escaped, you may use {!! !!}