HTML CSS JavaScript TypeScript PHP Tailwind Bootstrap Foundation Bulma Rec

Laravel | Blade Templates Inheritance

Last Updated : 22 Nov, 2019

Read Discuss Courses Practice Video

A templating engine makes writing frontend code easier and helps in reusing the code. All the blade files have a extension of **.blade.php*. In Laravel, most of the times frontend files are stored in *resources/views* directory. Blade files support PHP and are compiled into plain PHP and cached in the server so that we don't have to do the extra work of compiling the templates again when a user access a page again, thus using Blade is as efficient as using PHP files itself in the frontend.

Template Inheritance: In most of the modern webpages, a fixed theme is followed in all the webpages. Thus it is greatly effective to be able to reuse your code so that you don't have to write again the repeating parts in your code and Blade greatly helps you in achieving this.

- **Defining a layout:** Let's do that with an example and create a file called **layout.blade.php** in **resources/views** directory as shown below:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>@yield('title')</title>
</head>
<body>
    <div>
        @yield('content')
    </div>
</body>
</html>
```

one. These names are going to be used later in the child page to tell that this section is extended here.

- **Extending a layout:** Let's do that too now and create a page at **resources/views** directory called *mypage.blade.php* as given below:

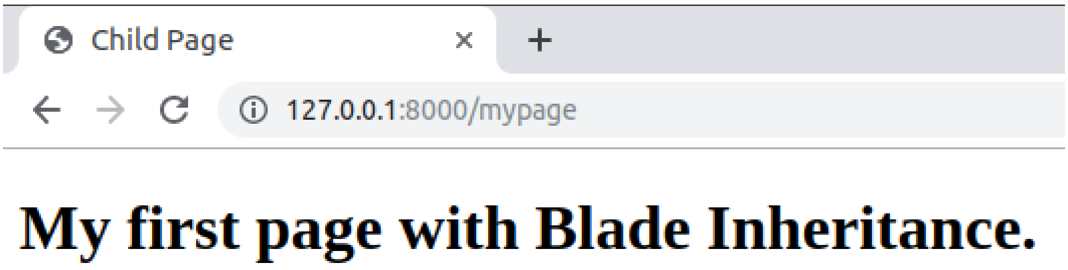
```
@extends('layout')

@section('title')
    Child Page
@endsection

@section('content')
    <h1>My first page with Blade Inheritance.</h1>
@endsection
```

In this code, we are first using the **@extends** directive which tells which blade page we are inheriting this page from. In our case, it is going to be layout as we are going to inherit this page from **layout.blade.php**, we created earlier. Further, we use the **@section** directive to extend each of the **@yield** directive's of the parent blade file. We have to tell the name of each **@yield** directive we are extending here in the **@section** directive as we have done in code above. Make sure after writing the code you end the directive with *@endsection*. All the **@yield** sections will be replaced with the respective code in the child blade pages. One last thing left to make this work is adding a route as given below in your **routes/web.php**.

```
Route::get('/mypage', function() {
    return view('mypage');
});
```

We just created a route to */mypage* and in the callback function we are serving *mypage.blade.php*. Notice that Blade automatically looks for files in *resources/views* directory.

- **Output:**



My first page with Blade Inheritance.

In the output you can see how **@yield('title')** is replaced with **Child Page** and **@yield('content')** is replaced with **My first page with Blade Inheritance.**

Related Articles

1. Laravel | Artisan Commands to know in Laravel
2. Laravel | Validation Rules
3. Laravel | Artisan Console Introduction
4. Laravel | Installation and Configuration
5. What are the differences and Similarities Between Lumen and Laravel?
6. Laravel | Directory Structure
7. Laravel | CSRF Protection
8. Laravel | Route::resource vs Route::controller
9. Different ways for passing data to view in Laravel

Like 1

Next

Laravel | Artisan Commands to know in Laravel

Article Contributed By :

gurrrung
@gurrrung

Vote for difficulty

Easy Normal Medium Hard Expert

Article Tags : Laravel, PHP, Web Technologies

Practice Tags : PHP

Improve Article Report Issue

GeeksforGeeks
A-143, 9th Floor, Sovereign Corporate Tower,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org