

Laravel - Blade Templates

Topics : [Laravel](#)

Written on [December 25, 2023](#)

Laravel's Blade templating engine is a powerful and intuitive tool for working with PHP views. Blade provides a convenient and clean syntax for common PHP tasks, and it also adds some additional features to make your views more expressive and readable.

Here are some key features of Blade templates in Laravel:

1. Echoing Data:

You can echo data in Blade using the `{{ }}` syntax:

```
<p>{{ $variable }}</p>
```

2. Raw Output:

If you need to display raw, unescaped data, you can use the `{!! !!}` syntax:

```
<p>{!! $rawHtml !!}</p>
```

3. Blade Comments:

You can add comments to your Blade views using the `{{-- --}}` syntax:

```
{{-- This is a Blade comment --}}
```

4. Control Structures:

Blade supports common control structures like `if`, `else`, `elseif`, `foreach`, and `forelse`:

```
@if($condition)
    // Code to execute if condition is true
@elseif($anotherCondition)
    // Code to execute if anotherCondition is true
@else
    // Code to execute if none of the conditions are true
@endif
```

```
@foreach($items as $item)
    // Loop through items
@endforeach
```

5. Include & Extend:

Blade allows you to include other Blade views using `@include` and create layouts using `@extends`:

```
@extends('layouts.master')

@section('content')
    // Content of the view
@endsection
```

6. Stacks:

Blade provides a convenient way to push to and pop from named stacks:

```
@push('scripts')
    // Push JavaScript to the 'scripts' stack
@endpush

@stack('scripts')
```

7. Component & Slots:

Laravel 7 and above introduced Blade components and slots, allowing you to create reusable components:

```
<!-- Alert Component -->
<x-alert>
    <x-slot name="title">Alert Title</x-slot>

    // Alert content
</x-alert>
```

8. Service Injection:

You can inject services directly into your views:

```
@inject('service', 'App\Services\SomeService')

<p>{{ $service->method() }}</p>
```

```
LARAVEL
├── app
├── bootstrap
├── config
├── database
├── lang
├── public
├── resources
│   ├── css
│   ├── js
│   └── views
│       └── welcome.blade.php
├── routes
├── storage
├── tests
├── vendor
├── .editorconfig
├── .env
├── .env.example
├── .gitattributes
├── .gitignore
├── artisan
├── composer.json
├── composer.lock
├── package.json
├── phpunit.xml
└── ...

resources > views > welcome.blade.php
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6
7     <title>Laravel</title>
8
9     <!-- Fonts -->
10    <link href="https://fonts.bunny.net/css2?family=Nunito:wght@400;600;700&display=swap" rel="stylesheet">
11
12    <!-- Styles -->
13    <style>
14        /*! normalize.css v8.0.1 | MIT License | github.com/necolas/normalize.css */html{line-height:1.15;-ms-text-size-adjust:100%;text-rendering:optimizeLegibility;}body{margin:0;font-family:'Nunito', sans-serif;}
15    </style>
16
17    <style>
18        body {
19            font-family: 'Nunito', sans-serif;
20        }
21    </style>
22 </head>
23 <body class="antialiased">
24     <div class="relative flex items-top justify-center min-h-screen bg-gray-100 dark:bg-gray-900">
25         @if (Route::has('login'))
26             <div class="hidden fixed top-0 right-0 px-6 py-4 sm:block">
27                 @auth
28                     <a href="{{ url('/home') }}" class="text-sm text-gray-700 dark:text-gray-500">Home</a>
29                 @else
30                     <a href="{{ route('login') }}" class="text-sm text-gray-700 dark:text-gray-500">Log in</a>
31                 @endauth
32             </div>
33         @endif
34     </div>
35 </body>
36 </html>
```

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)