Blade Templates & Partials

One of the biggest benefits of a templating language is the ability to create reusable layouts and partials. Jigsaw gives you access to all the templating features and control structures of Blade that are available in Laravel (learn more about Blade layouts in the official Blade documentation).

Defining a Layout

Layouts themselves are just basic Blade templates that have one or more @yield calls where child templates can render their contents.

A basic master layout could look like this:

```
<!DOCTYPE html>
<html>
  <head>
    <title>The Amazing Web</title>
  </head>
  <body>
    <header>
      My Amazing Site
    </header>

    @yield('content')

    <footer>
      <p>©{{ date('Y') }} Awesome Co</p>
    </footer>
  </body>
</html>
```

Jigsaw provides a /source/_layouts directory out of the box with a basic master layout.

Extending a Layout

To extend a layout, create a template that specifies which layout to extend in an @extends directive, and which section(s) to populate using the @section directive:

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

@section('content')
  <div>
    <p>The contents of my amazing homepage.</p>
  </div>
@endsection
```

Layouts and partials are referenced relative to the source directory using dot notation, where each dot represents a directory separator in the file name and the .blade.php extension omitted.

Partials

To include a template inside of another template, use the @include directive:

```
<!DOCTYPE html>
<html>
  <head>
    <title>The Amazing Web</title>
  </head>
  <body>
    @include('_partials.header')

    @yield('content')

     @include('_partials.footer')
  </body>
</html>
```

You can pass data to a partial by passing an associative array as a second parameter:

```
<!DOCTYPE html>
<html>
  <head>
    <title>The Amazing Web</title>
  </head>
  <body>
    @include('_partials.header', ['page_title' => 'My Amazing Site'])

    @yield('content')

     @include('_partials.footer')
  </body>
</html>
```

That data is then available in your partial as a normal variable:

```
<!-- _partials/header.blade.php -->
<header>
  {{ $page_title }}
</header>
```

Components

Jigsaw supports both class based and anonymous Blade components.

To display a component, you may use a Blade component tag within one of your Blade templates. Blade component tags start with the string x- followed by the kebab case name of the component class:

1<x-input />
In Jigsaw, views are auto-discovered from the source/_components directory; to create an anonymous <x--style components, you only need to place a Blade template within that directory.

Class-based components can be manually registered using $bladeCompiler->component(), as detailed in the Extending Blade with custom directives section below; or, they can be auto-discovered by using the Components namespace. To autoload class-based components that use the Components namespace, add an autoload entry to your composer.json file:

composer.json

```
1"autoload": {
2    "psr-4": {
3        "Components\\": "where/you/want/to/keep/component/classes/"
4    }
5}
```

... and then update Composer's autoload references by running composer dump-autoload in your terminal.

Preventing layouts, partials & components from rendering
Since it's important that layouts, partials and components are never rendered on their own, you need to be able to tell Jigsaw when a file shouldn't be rendered.

To prevent a file or folder from being rendered, simply prefix it with an underscore:

source
_assets
_layouts
master.blade.php
assets
index.blade.php
...
Jigsaw gives you a /_layouts directory by default, but you can create any files or directories you need; anything prefixed with an underscore will not be rendered directly to /build_local.

For example, if you wanted a place to store all of your partials, you could create a directory called _partials:

```
source
_assets
_layouts
_partials
footer.blade.php
header.blade.php
assets
index.blade.php
...
```

Since the _partials directory starts with an underscore, those files won't be rendered when you generate your site, but will still be available to @include in your other templates.

Extending Blade with custom directives

Jigsaw gives you the ability to extend Blade with custom directives, just as you can with Laravel. To do this, create a blade.php file at the root level of your Jigsaw project (at the same level as config.php), and return an array of directives keyed by the directive name, each returning a closure.

For example, you can create a custom @datetime($timestamp) directive to format a given integer timestamp as a date in your Blade templates:

blade.php

```
return [
    'datetime' => function ($timestamp) {
        return '<?php echo date("l, F j, Y", ' . $timestamp . '); ?>';
    }
];
```

Alternatively, the blade.php file receives a variable named $bladeCompiler, which exposes an instance of \Illuminate\View\Compilers\BladeCompiler. With this, you can create custom Blade directives, aliased components, named @include statements, or other extended Blade control structures:

blade.php

```
<?php

/** @var \Illuminate\View\Compilers\BladeCompiler $bladeCompiler */

$bladeCompiler->directive('datetime', function ($timestamp) {
```

```
    return '<?php echo date("l, F j, Y", ' . $timestamp . '); ?>';
});

$bladeCompiler->aliasComponent('_components.alert');

$bladeCompiler->include('includes.copyright');
```

page.blade.php

```
<!-- Before -->

@component('_components.alert')
    Pay attention to this!
@endcomponent

@include('_partials.meta.copyright', ['year' => '2018'])

<!-- After -->

@alert
    Pay attention to this!
@endalert

@copyright(['year' => '2018'])
```

Specifying Blade hint paths

To use Blade hint paths/namespaces in your markup (for example, email:components::section), specify the path to the directory using the viewHintPaths key in config.php:

config.php

```
<?php

return [
    'viewHintPaths' => [
        'email:templates' => __DIR__.'/source/_layouts',
        'email:components' => __DIR__.'/source/_components',
        'email:partials' => __DIR__.'/source/_partials'
    ]
];
```