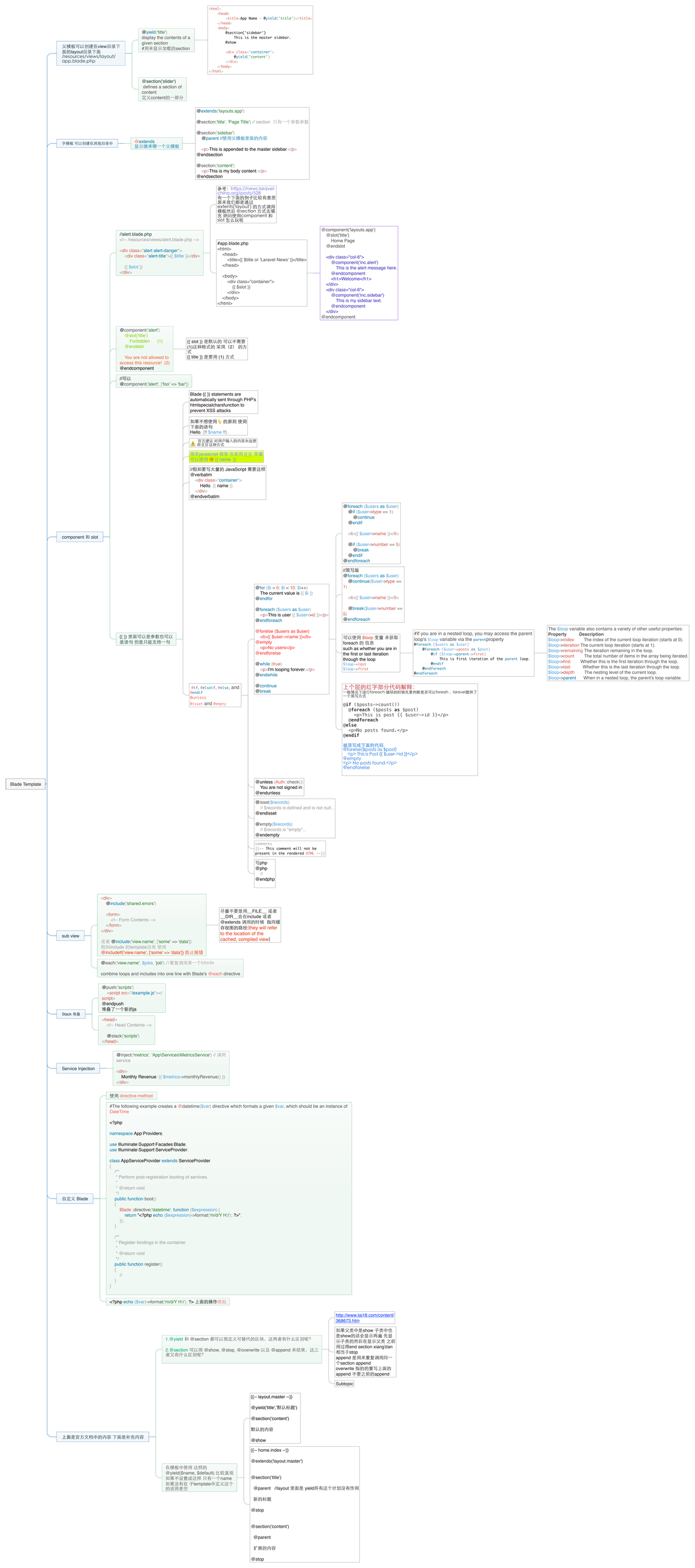
Blade Template
component 和 slot
父模板可以创建在view目录下面的layout目录下面 /resources/views/layout/app.blade.php
@yield('title')
display the contents of a given section
#用来显示加载的section
<html>
<head>
<title>App Name - @yield('title')</title>
</head>
<body>
@section('sidebar')
This is the master sidebar.
@show

@yield('content')

</body>
</html>
@section('slider')
defines a section of content
定义content的一部分
字模板 可以创建在其他目录中
@extends
显示继承哪一个父模板
@extends('layouts.app')
@section('title', 'Page Title') // section 只有一个参数参数
@section('sidebar')
@parent //使用父模板里面的内容
This is appended to the master sidebar.
@endsection
@section('content')
This is my body content.
@endsection
参考： https://news.laravel-china.org/posts/528
有一个下面的例子比较有意思 原来我们都是通过 extends('layout') 的方式调用模板然后 和section 方式去填充 纳闷使用component 和 slot 怎么玩呢
#app.blade.php
<html>
<head>
<title>{{ $title or 'Laravel News' }}</title>
</head>
<body>

{{ $slot }}

</body>
</html>
@component('layouts.app')
@slot('title')
Home Page
@endslot

@component('inc.alert')
This is the alert message here.
@endcomponent
<h1>Welcome</h1>

@component('inc.sidebar')
This is my sidebar text.
@endcomponent

@endcomponent
//alert.blade.php
<!-- /resources/views/alert.blade.php -->

{{ $title }}
{{ $slot }}

@component('alert')
@slot('title')
Forbidden (1)
@endslot
You are not allowed to access this resource! (2)
@endcomponent
{{ slot }} 是默认的 可以不需要 (1)这种格式的 采用 (2) 的方式
{{ title }} 是要用 (1) 方式
//可以
@component('alert', ['foo' => 'bar'])
{{ }} 里面可以是参数也可以是语句 但是只能支持一句
Blade {{ }} statements are automatically sent through PHP's htmlspecialcharsfunction to prevent XSS attacks
如果不想使用👆 的原则 使用下面的语句
Hello, !!! $name !!!.
⚠️ 官方建议 对用户输入的内容永远使用{{ }}这种方式
但是javascript 框架 也是用 {{ }} 变量 所以冲突 使用 @ if name 为
//假如要写大量的 JavaScript 需要这样
@verbatim

Hello, {{ name }}.

@endverbatim
@if, @elseif, @else, and @endif
@unless
@isset and @empty
@for ($i = 0; $i < 10; $i++)
The current value is {{ $i }}
@endfor
@foreach ($users as $user)
This is user {{ $user->id }}
@endforeach
@forelse ($users as $user)
{{ $user->name }}
@empty
No users
@endforelse
@while (true)
I'm looping forever.
@endwhile
@continue
@break
@foreach ($users as $user)
@if ($user->type == 1)
@continue
@endif
{{ $user->name }}
@if ($user->number == 5)
@break
@endif
@endforeach
//简写版
@foreach ($users as $user)
@continue($user->type == 1)
{{ $user->name }}
@break($user->number == 5)
@endforeach
可以使用 $loop 变量 来获取 foreach 的 信息
such as whether you are in the first or last iteration through the loop
$loop->last
$loop->first
#If you are in a nested loop, you may access the parent loop's $loop variable via the parent property
@foreach ($users as $user)
@foreach ($user->posts as $post)
@if ($loop->parent->first)
This is first iteration of the parent loop.
@endif
@endforeach
@endforeach
The $loop variable also contains a variety of other useful properties:
Property Description
$loop->index The index of the current loop iteration (starts at 0).
$loop->iteration The current loop iteration (starts at 1).
$loop->remaining The iteration remaining in the loop.
$loop->count The total number of items in the array being iterated.
$loop->first Whether this is the first iteration through the loop.
$loop->last Whether this is the last iteration through the loop.
$loop->depth The nesting level of the current loop.
$loop->parent When in a nested loop, the parent's loop variable.
上个层的红字部分代码解释：
一般情况下进行foreach 循环的时候先要判断是否可以foreah ，laravel提供了一个便写方式
@if ($posts->count())
@foreach ($posts as $post)
This is post {{ $user->id }}
@endforeach
@else
No posts found.
@endif
被简写成下面的代码
@forelse($posts as $post)
 This is Post {{ $user->id }}
@empty
 No posts found.
@endforelse
@unless (Auth::check())
You are not signed in
@endunless
@isset($records)
// $records is defined and is not null...
@endisset
@empty($records)
// $records is "empty"...
@endempty
comment
{{-- This comment will not be present in the rendered HTML --}}
写php
@php
//
@endphp
sub view

@include('shared.errors')
<form>
<!-- Form Contents -->
</form>

或者 @include('view.name', ['some' => 'data'])
假如include 的template没有 使用
@includeIf('view.name', ['some' => 'data']) 防止报错
尽量不要是用__FILE__ 或者 __DIR__ 会在include 或者 @extends 调用的时候 指向硬存在视图的路径(they will refer to the location of the cached, compiled view)
@each('view.name', $jobs, 'job') //重复调用某一个blade
combine loops and includes into one line with Blade's @each directive
Stack 堆叠
@push('scripts')
<script src="/example.js"></script>
@endpush
堆叠了一个新的js
<head>
<!-- Head Contents -->
@stack('scripts')
</head>
Service Injection
@inject('metrics', 'App\Services\MetricsService') // 调用service

Monthly Revenue: {{ $metrics->monthlyRevenue() }}.

自定义 Blade
使用 directive method
#The following example creates a @datetime($var) directive which formats a given $var, which should be an instance of DateTime
<?php
namespace App\Providers;
use Illuminate\Support\Facades\Blade;
use Illuminate\Support\ServiceProvider;
class AppServiceProvider extends ServiceProvider
{
/**
* Perform post-registration booting of services.
*
* @return void
*/
public function boot()
{
Blade::directive('datetime', function ($expression) {
return "<?php echo ($expression)->format('m/d/Y H:i'); ?>";
});
}
/**
* Register bindings in the container.
*
* @return void
*/
public function register()
{
//
}
}
<?php echo ($var)->format('m/d/Y H:i'); ?> 上面的操作类似
上面是官方文档中的内容 下面是补充内容
1.@yield 和 @section 都可以预定义可替代的区块，这两者有什么区别呢？
2.@section 可以用 @show, @stop, @overwrite 以及 @append 来结束，这三者又有什么区别呢？
http://www.lai18.com/content/368673.htm
如果父类中是show 子类中也是show的话会显示两遍 先显示子类的然后在显示父类 之前用过得end section xiang'dan 相当于stop
append 是用来重复调用同一个section append
overwrite 指的的重写上面的append 不要之前的append
Subtopic
在模板中使用 这样的 @yield('$name, $default) 比较直观 如果不设置成这样 只有一个name 如果没有在 子template中定义这个的话将是空
{{-- layout.master --}}
@yield('title','默认标题')
@section('content')
默认的内容
@show
{{-- home.index --}}
@extends('layout.master')
@section('title')
@parent //layout 里面是 yield所有这个计划没有作用
新的标题
@stop
@section('content')
@parent
扩展的内容
@stop