WCBA2019

HOW TO WORK WITH WP IN A MODERN WAY: THEMOSIS

PAVOL CABAN

Developer @Webikon

UNDERSCORES STARTER THEME

underscores.me

- Predefined files structure, helpers, basic JS/CSS
- Later SASS, Gulp

TIMBER

www.upstatement.com/timber

- WP Plugin
- Twig, object-oriented patterns and MVC principles

```php
<?php
// Old way
$thumb_id = get_post_thumbnail_id($post->ID);
$url = wp_get_attachment_url($thumb_id);
?>
<img src="<?php echo $url; ?>" alt="Thumbnail for <?php echo $post->post_title; ?>" />
```

```twig
// Timber way
<img src="{{ post.thumbnail.src }}" alt="Thumbnail for {{ post.title }}" />
```

SAGE STARTER THEME

roots.io/sage

- Namespaces
- Controllers
- Blade templates
- Webpack
- Different files structure

SAGE THEME STRUCTURE

```
themes/your-theme-name/   # → Root of your Sage based theme
├── app/                  # → Theme PHP
│   ├── controllers/      # → Controller files
│   ├── admin.php         # → Theme customizer setup
│   ├── filters.php       # → Theme filters
│   ├── helpers.php       # → Helper functions
│   └── setup.php         # → Theme setup
├── config/               # → Theme configuration
├── composer.json         # → Autoloading for `app/` files
├── composer.lock         # → Composer lock file (never edit)
├── dist/                 # → Built theme assets (never edit)
├── node_modules/         # → Node.js packages (never edit)
├── package.json          # → Node.js dependencies and scripts
├── resources/            # → Theme assets and templates
│   ├── assets/           # → Front-end assets
│   │   ├── config.json   # → Settings for compiled assets
│   │   ├── build/        # → Webpack and ESLint config
│   │   ├── fonts/        # → Theme fonts
│   │   ├── images/       # → Theme images
│   │   ├── scripts/      # → Theme JS
│   │   └── styles/       # → Theme stylesheets
│   ├── functions.php     # → Composer autoloader, theme includes
│   ├── index.php         # → Never manually edit
│   ├── screenshot.png    # → Theme screenshot for WP admin
│   ├── style.css         # → Theme meta information
│   └── views/            # → Theme templates
│       ├── layouts/      # → Base templates
│       └── partials/     # → Partial templates
└── vendor/               # → Composer packages (never edit)
```

THEMOSIS

framework.themosis.com

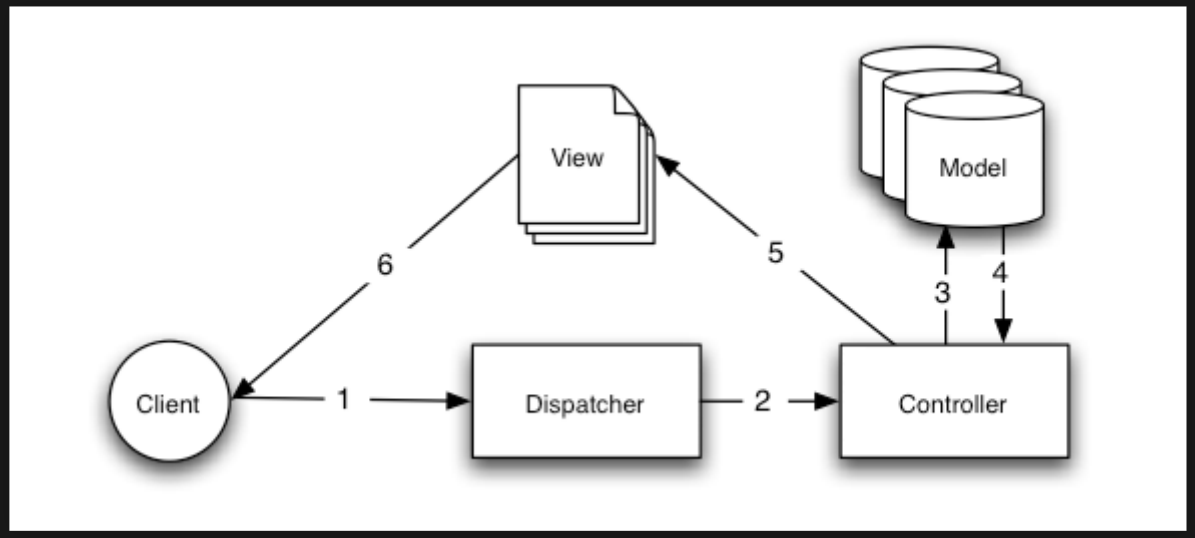
- Models
- **Controllers**
- Routing
- **Blade Templates**
- Application Structure
- Namespaces
- Composer

THEMOSIS APPLICATION STRUCTURE

```
├── app/
│   ├── Console
│   ├── Exceptions
│   ├── Forms
│   ├── Hooks
│   ├── Http
│   │   ├── Controllers
│   ├── Mail
│   ├── Providers
│   ├── Widgets
├── bootstrap/
├── config/
│   ├── app.php
│   ├── ...
│   ├── wordpress.php
├── database/
├── htdocs/
│   ├── cms/
│   ├── content/
│   ├── index.php
│   ├── wp-config.php
├── resources/
│   ├── languages/
│   ├── views/
├── routes/
│   ├── console.php
│   ├── web.php
├── storage/
├── tests/
├── vendor/
├── .env
├── composer.json
├── console
├── wp-cli.yml
```

MODEL VIEW CONTROLLER

- **Model** - database layer
- **View** - UI display
- **Controller** - data flow



WHY MVC?

- Separation of concerns - data logic and frontend templates
- Better data flow control
- Modularity and better structure
- Maintainability

CONTROLLERS AND WORKING WITH DATA

Principles:

- separate data logic from templates
- use Controllers to pass all data to templates
- avoid using WP functions in templates
- if necessary, use helper functions to get additional data
- keep it DRY

HOMEPAGE CONTROLLER EXAMPLE

```php
namespace App\Http\Controllers;

class HomepageController extends Controller
{
    public function front(\WP_Post $post)
    {
        return view('pages.front-page', [
            'page_data' => [
                'title' => get_the_title($post->ID),

                'hero_data' => [
                    'title' => get_post_meta($post->ID, 'hero_title', true),
                    'text' => get_post_meta($post->ID, 'hero_text', true),
                ],
            ]
        ]);
    }
}
```

```php
// Route for Homepage
Route::any('/', 'HomepageController@front');
```

BLADE TEMPLATES

laravel.com/docs/5.7/blade

TEMPLATE INHERITANCE & EXTENDING A LAYOUT

```
<!-- Stored in htdocs/content/themes/theme_slug/views/_layouts/app.blade.php -->
<html>
    <head>
        <title>WCBA 2019</title>
    </head>

    <body>
        <div class="container">
            <!-- This section will be changed -->
            @yield('page_content')
        </div>
    </body>
</html>
```

```
<!-- Stored in htdocs/content/themes/theme_slug/views/pages/front-page.blade.php -->
@extends('_layouts.app')

@section('page_content')
    <p>This is my homepage content.</p>
@endsection
```

```
<!-- Stored in htdocs/content/themes/theme_slug/views/blog/single-post.blade.php -->
@extends('_layouts.app')

@section('page_content')
    <h1>Post title</h1>

    <p>Lorem ipsum....</p>
@endsection
```

REGULAR TEMPLATE VS BLADE

Single Post

```php
<?php
get_header();
?>
    <div id="primary" class="content-area">
        <main id="main" class="site-main">
            <?php
            while (have_posts()) : the_post();

                the_content();

            endwhile;
            ?>
        </main>
    </div>
<?php
get_footer();
?>
```

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

@section('page_content')
    @while(have_posts()) @php the_post() @endphp

        @php the_content() @endphp

    @endwhile
@endsection
```

INCLUDING SUB-VIEWS INTO VIEWS

```
<!-- Stored in htdocs/content/themes/theme_slug/views/pages/front-page.blade.php -->
@extends('_layouts.app')

@section('page_content')
    <p>This is my homepage content.</p>

    <!-- Include static Hero component -->
    @include('_components.hero')
@endsection
```

```
<!-- Stored in htdocs/content/themes/theme_slug/views/_components/hero.blade.php -->
<div class="hero">
    <h2 class="hero__title">
        Title
    </h2>

    <div class="hero__text">
        Some text
    </div>
</div>
```

PASSING DATA TO VIEWS

```
<!-- Stored in htdocs/content/themes/theme_slug/views/pages/front-page.blade.php -->
@extends('_layouts.app')

@section('page_content')
    <p>This is my homepage content.</p>

    <!-- Include Hero component and pass it some data -->
    @include('_components.hero', [
        'data' => [
            'title' => 'Hero title',
            'text' => 'Hero text',
        ]
    ])
@endsection
```

```
<!-- Stored in htdocs/content/themes/theme_slug/views/_components/hero.blade.php -->
<div class="hero">
    <h2 class="hero__title">
        {{ $data['title'] }}
    </h2>

    @isset ($data['text'])
        <div class="hero__text">
            {!! $data['text'] !!}
        </div>
    @endisset
</div>
```

SECURITY

Automatic escaping with `{{ }}` using `htmlspecialchars` to prevent XSS

```
// Escape this
Hello, {{ $name }}.

// Don't escape this
Hello, {!! $name !!}.
```

PERFORMANCE

All blade templates are compiled and cached.

POSSIBILITY TO USE PHP

```
<!-- Stored in htdocs/content/themes/theme_slug/views/pages/front-page.blade.php -->
@extends('_layouts.app')

@section('content')
    <p>This is my homepage content.</p>

    @include('_components.hero')

    <?php
    $var = 1;

    echo $var;
    echo 'I can also use PHP in Blade';
    ?>
@endsection
```

TRY IT YOURSELF

Requirements:

- Localhost with PHP >= 7.1 (7.2 recommended)
- Composer
- Node.js >= 8.0.0

- How to install Sage
- How to install Themosis

Tip:

Use Brew (MacOS) or Chocolatey (Windows) to install Composer, Node.js or some other required libraries

```
brew install composer
brew install node@8
```

CAVEATS

- takes an extra time to learn and customize
- issues with some plugins
  - Woocommerce

ADDITIONAL LINKS

- github.com/Webikon/_s-foundation-sites
  (our customized Underscores)
- github.com/Webikon/timber-starter-theme
  (our starter theme for Timber)
- github.com/Webikon/sage-starter-project
  (our starter project for Sage)
- github.com/Webikon/themosis-starter-project
  (our Themosis starter project)
- github.com/soberwp/controller
  (Controller package used in Sage)
- github.com/themosis/bookstore/tree/2.0
  (example Themosis project)

THANK YOU

facebook: facebook.com/pavol.caban.3
twitter: @pavol_caban
linkedin: linkedin.com/in/pavolcaban

Also, we are hiring!
hello@webikon.sk