Sage: A Modern Developer’s Starter Theme

Julien Melissas

WordCamp Atlanta 2015

Hello, I’m Julien.

@JulienMelissas

www.julienmelissas.com

www.craftpeak.com

www.jmlabs.io

Things I like to use…

- Sublime Text Editor
- Git (SourceTree GUI) for version control
- Twitter Bootstrap
- CSS Preprocessors
- Vagrant (local development)

Roots / roots.io / @rootswp

- Projects that help you build better WordPress sites.
- A group of awesome people who want to help bring modern development practices to WordPress.

Projects

Sage

Bedrock

What is Sage?

Hit the ground running. Delete key friendly.

When use Sage?

- Brand new *custom* theme
- Quick, responsive site or application
- Want to sharpen your skills, with real-world example (not just starting from scratch)

Who should use Sage?

- Theme developers who want a reliable toolkit they don’t want to personally maintain
- People getting into more advanced theme development
- People who need a working but delete-key friendly starter theme
- http://roots.io/roots-should-not-be-your-first-wordpress-theme/

Disclaimers/Downsides

- Learning curve
- Potential theme-wrapper conflicts with some other plugins/frameworks
- Opinionated (it’s everything though?)
- Development moves fast/can be difficult to update
- Not coded to exact “WordPress coding standards”, uses PSR-2 for formatting, higher PHP requirements

Sage Features

- Advanced build workflow (with gulp, Bower and asset-builder) to minify scripts, styles (Sass and Less), and images. Uses BrowserSync.
- Theme Wrapper
- HTML5 Boilerplate, microformats, Bootstrap included
- Multilingual ready & 30+ translations

Prerequisites

- PHP >= 5.4
- Node/NPM Installed: http://nodejs.org/ (Installing Node via homebrew or NVM is recommended)
- gulp.js: http://gulpjs.com/
- Bower: http://bower.io/

Installing Sage

- Our instructions are actually pretty good: https://github.com/roots/sage#installation
- Set WP_ENV
- Download zip/git clone repo
- Run `` `npm install` ``
- Change `devUrl` in `manifest.json` to your development hostname
- Run `gulp` and then `` `gulp watch` ``

The Wrapper

- Reduces extra calls for get_header(), get_footer(), get_sidebar(), keeps things DRY
- Follows/extends template hierarchy
- Easy to customize single page template designs
- Read more: https://roots.io/sage/docs/theme-wrapper/

The /lib/ folder

- `config.php`: Configure Sidebars / Google Analytics
- `nav.php`: rewrites WordPress default Walker Class to work with Bootstrap-specific navbar markup
- `scripts.php`: calls the correct assets for the page
- `titles.php`: powers the logic behind the roots `title();` function

Why >= 5.4?

- Namespacing & more info: https://roots.io/upping-php-requirements-in-your-wordpress-themes-and-plugins/
- Short Echo Syntax: <?= ‘hello world’ ?> instead of <?php echo ‘hello world’ ?>
- Short array syntax: $array = [1, 2, 3, 4]; instead of $array = array(1, 2, 3, 4);

Project Assets

- gulp: build tool written in JavaScript to take care of those tasks that we need to do every time we hit “save”

- Commands:

  - `gulp`: compiles development CSS and JS

  - `gulp watch`: watches for code changes, performs `gulp`

  - gulp --production: compiles production assets (and manifest.json)

Assets: Bower & asset-builder

- Bower: package manager for front-end tools. Used for pulling in/downloading required assets like jQuery, Bootstrap, and others you can add yourself.

- asset-builder: reads project's `manifest.json` file and automatically puts all your assets in the right place.

Assets: Editing CSS

- All files funnel into `main.less` or `main.sass`
- `_variables.less` overrides Bootstrap variables, but also good for you to add your own variables
- `/components/` folder is for re-usable elements in your site
- `/layouts/` folder is for common layout elements, and `/pages/` folder is for page-specific styles

Demo BrowserSync

Because live reloading is just not cool enough.

Assets: Editing JS

- _main.js file uses DOM-based routing based on a post from Paul Irish’s blog - has you use the body class to target code that will fire on that specific page

- If body class contains ‘-‘, replace it with ‘_’ (for example `.about-us` would become `about_us`

Deployment Methods

- Bedrock & Capistrano
- Git deploy service (like Beanstalk or FTPloy) - requires you to check in your assets unless you use a deployment hook.
- `` `gulp --production` `` and good ol’ FTP

Learning More/Getting Help

- Site/Blog: http://roots.io (has some learning resources)
- Dedicated forum: http://discourse.roots.io
- Ping me!
- Contributions are always welcome on GitHub, we’re always making improvements and everyone’s opinion counts.

The Future of Sage

- Yeoman: “Yeoman helps you to kickstart new projects, prescribing best practices and tools to help you stay productive.”

- One command: `yo sage` will generate a starter theme customized for your project.

- Framework agnostic: Bootstrap, Foundation, or nothing.

- Preprocessor agnostic: Less or Sass

Demo `yo sage`

This is exclusive. You must be at WordCamp ATL.

Thanks!

Any questions?