

# Introduction to the Juicebox Boilerplate

The Juicebox WordPress Boilerplate draws its inspiration from the Roots.io Bedrock framework (<https://roots.io/bedrock/>), and its file organization resembles that of placing WordPress in its subdirectory. However, the Boilerplate comes with several enhancements, including a superior folder structure that ensures sensitive files remain shielded from the web root. Specifically, the necessary public files are housed in the `www/` directory, while the `wp-content/` folder is renamed as `"app"` to reflect its contents better. Additionally, the `app` directory not only contains `"static content"` but also application code.

Another advantage of using the Juicebox WordPress Boilerplate is that it employs Dependency management through Composer, with Composer-managed dependencies being installed in the `vendor/` folder. WordPress configuration is simplified using environment-specific files and environment variables with Dotenv. Moreover, the Boilerplate comes with an autoloader for mu-plugins, where regular plugins/ can be utilized as must-use plugins mu-plugins/, ultimately improving the security of the framework.

## Composer

The Juicebox WordPress Boilerplate utilizes Composer (<https://getcomposer.org/>) to manage dependencies for PHP. The Boilerplate deems any third-party library, including WordPress and its associated plugins, as a dependency. As such, it is imperative to download and install these packages before embarking on any website development or deploying it to the server.

## Plugins

The Juicebox WordPress Boilerplate leverages WordPress Packagist (<http://wpackagist.org/>) to acquire any plugins that are obtainable on the WordPress Plugin Directory (<https://wordpress.org/plugins/>). By default, the contents of the `plugins` and `mu-plugins` folders are Git ignored since they are managed by Composer. If you need to incorporate a plugin that is unavailable via Composer/Packagist, you will have to modify `.gitignore` to include them. For more detailed information, please consult the "Managing Plugins" section.

## Theme

The WordPress theme is constructed using Timber & Twig, with Timber utilizing the Twig syntax to render templates. To gain a better understanding of how both Timber and Twig operate within WordPress, you may refer to further reading materials available on CSS Tricks (<https://css-tricks.com/timber-and-twig-reignited-my-love-for-wordpress/>).

The theme encompasses a wide range of modules and components that rely heavily on Advanced Custom Fields (ACF <https://www.advancedcustomfields.com/>) to personalize the theme's appearance and functionality.

Moreover, the theme utilizes Node Package Manager (NPM <https://www.npmjs.com/>) to manage third-party JavaScript libraries, while Node (<https://nodejs.org/en/>) is leveraged to compile, concatenate, and minify JavaScript and SCSS files to CSS. For additional details, please refer to the "Theme Development" section.