Laranav

Release

April 03, 2016

Contents

1 User Guide **3**

1.1 Overview . . . 3
1.1.1 Requirements . . . 3
1.1.2 Installation . . . 3
1.1.3 License . . . 3
1.2 Quickstart . . . 4
1.2.1 Creating a menu . . . 4
1.3 FAQ . . . 4
1.3.1 Why is (insert-component-here) not working? . . . 4
1.4 Developers . . . 5
1.4.1 Contributing . . . 5

An easy-to-use, customisable menu and navigation package, for Laravel 5.1+. It offers:

- Define your menus in simple config files
- Use regular ol' Blade templates for rendering menus - never build HTML programatically again!
- Fully-tested

CHAPTER 1

User Guide

1.1 Overview

1.1.1 Requirements

1. PHP 5.5.9+
2. Laravel 5.1+

1.1.2 Installation

The recommended way to install Laranav is with Composer. Composer is a dependency management tool for PHP that allows you to declare the dependencies your project needs and installs them into your project.

```
# Install Composer
curl -sS https://getcomposer.org/installer | php
```

You can add Laranav as a dependency using the `composer.phar` CLI:

```
php composer.phar require larabros/laranav:~1.0
```

Alternatively, you can specify it as a dependency in your project's existing `composer.json` file:

```
{
  "require": {
     "larabros/laranav": "~1.0"
  }
}
```

Then, open `config/app.php` in your Laravel project and add the service provider. You can optionally add the `Nav` facade:

After registering the service provider, run `php artisan vendor:publish` to publish Laranav's config files and example template files to your project.

1.1.3 License

Licensed using the MIT license.

Copyright (c) 2016 Hassan Khan <contact@hassankhan.me>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2 Quickstart

This page provides a quick introduction to Laranav and introductory examples. If you have not already installed, head over to the *Installation* page.

1.2.1 Creating a menu

Configuration

First, open `config/laranav/config.php`:

Any menus added here automatically inherit their configuration from `default`, and can be overridden by specifying them in explicitly for each menu. You can read more about available options [here](#configphp).

Menu items

Next, we will define items for the menu in `config/laranav/menus.php`:

Items can link to a simple URL, or you can use Laravel's *UrlGenerator* object to generate URLs. You can read more about available options [here](#menusphp)

Rendering the menu

Finally, in your template, add *{!! Nav::menu('myMenu')->toHtml() !!}* and your menu should render!

1.3 FAQ

1.3.1 Why is (insert-component-here) not working?

We're working on it, and we're trying to do this in a sane and sensible way, so it may take a while to get it right.

1.4 Developers

1.4.1 Contributing

Contributions are **welcome** and will be fully **credited**.

We accept contributions via Pull Requests on Github.

Documentation

Documentation for this project is available at http://laranav.readthedocs.org.

The API reference documentation is available at http://laranav.readthedocs.org/en/latest/Larabros/Laranav.

Pull Requests

- **PSR-2 Coding Standard:** The easiest way to apply the code standard is to install PHP Code Sniffer.
- **PHP 5.5.9:** Laranav has a minimum PHP version requirement of PHP 5.5.9. Pull requests must not require a PHP version greater than PHP 5.5.9 unless the feature is only utilized conditionally.
- **Add tests!:** All pull requests must include unit tests to ensure the change works as expected and to prevent regressions.
- **Documentation format:** All documentation **except** this file, `CHANGELOG.md`, `LICENSE.md` and `README.md` are in reStructuredText - this includes any code docblocks. Using reST allows us to generate a better experience for users browsing the documentation.
- **Document any change in behaviour:** Make sure the `README.md` and any other relevant documentation are kept up-to-date.
- **Consider our release cycle:** We try to follow SemVer v2. Randomly breaking public APIs is not an option.
- **Use Git Flow:** Don't ask us to pull from your master branch. Set up Git Flow and create a new feature branch from `develop`.
- **One pull request per feature:** If you want to do more than one thing, send multiple pull requests.
- **Send coherent history:** Make sure each individual commit in your pull request is meaningful. If you had to make multiple intermediate commits while developing, please squash them before submitting.

Running tests

In order to contribute, you'll need to checkout the source from GitHub and install dependencies using Composer:

```
$ git clone https://github.com/larabros/laranav.git
$ cd laranav && composer install
$ php vendor/bin/phpunit
```

Reporting a security vulnerability

We want to ensure that Laranav is secure for everyone. If you've discovered a security vulnerability, we appreciate your help in disclosing it to us in a responsible manner.

Publicly disclosing a vulnerability can put the entire community at risk. If you've discovered a security concern, please email us at contact@hassankhan.me. We'll work with you to make sure that we understand the scope of the issue, and

that we fully address your concern. We consider correspondence sent to this email address our highest priority, and work to address any issues that arise as quickly as possible.

After a security vulnerability has been corrected, a security hotfix release will be deployed as soon as possible.

Happy coding!