
NukaCode - Menu Documentation

Release 1.0.4

RiDdLeS

March 10, 2016

1	Badges	3
1.1	Links	3
1.2	Guides	3

This package is designed to manage menus for your application.

1.1 Links

- [GitHub](#)
- [Packagist](#)

1.2 Guides

1.2.1 Installation

Installation guide for the NukeCode Menu package. This package requires Laravel 5.0 or higher.

Composer

Run the following command to require the menu package in your project.

```
composer require nukacode/menu:1.0.*
```

Service Provider

Add the following to the service providers in `config/app.php`.

```
'NukaCode\Menu\MenuServiceProvider',
```

Middleware

Add the following to the kernel in `app\kernel.php`

```
'menu' => '\NukaCode\Menu\Middleware\MenuMiddleware',
```

1.2.2 Quick Start

This is a short guide meant to get you up and running quickly.

Creating a new menu

When adding a new menu you need to specify the menu name. This is the name that will be used to access the menu object in the future.

```
$menu = \Menu::getMenu('leftMenu');
```

Adding a link to the menu

To add a link to the menu you just call the method link. This method takes a slug as the first parameter. This slug is used in insertAfter and insertBefore. The second parameter is a callback with your link data.

```
$leftMenu->link('home', function (Link $link) {  
    $link->name = 'The name of the link';  
    $link->url = 'A url or you can use the route method with a laravel named route';  
    $link->options[] = 'Add another option here';  
});
```

Adding a drop down to the menu

To add a drop down just add a drop down method. The first parameter is the slug. The second is the text for the drop down. The third is a call back where you will add your link methods.

```
$rightMenu->dropDown('user', \Auth::user()->username, function (DropDown $dropDown) {  
    $dropDown->link('profile.edit', function (Link $link) {  
        $link->name = 'Edit your profile';  
        $link->url = 'user/profile/';  
    });  
    $dropDown->link('logout', function (Link $link) {  
        $link->name = 'Logout';  
        $link->url = route('logout');  
    });  
});
```

Accessing the menu from your view or controller

In your view or controller just you will need to call render

```
$menu = \Menu::render('leftMenu');
```

This will return your menu object. From there you can loop through the links and add the appropriate html.

```
@foreach ($menu->link as $link)  
    <a href="{{ $link->url }}">{{ $link->name }}</a>  
@endforeach
```

1.2.3 Menus

A menu is a container for your links and drop downs. You can have as many menus as you want, The names just has to be unique. The name is used as the array key when getting a menu.

Creating a menu

To create a menu just call the `getMenu` method and give the menu a unique name.

```
\Menu::getMenu('menu name');
```

Checking the existence of a menu

You can check the existence of a menu before trying to retrieve it.

```
\Menu::exists('menu name');
```

This will return a boolean value with true meaning the menu exists.

Retrieving a menu

To retrieve a menu you just have to call the `render` method and supply the menu name.

```
\Menu::render('menu name');
```

This will return the Menu object that you configured.

Adding a link

Add a link to the menu.

Parameters

- slug: The slug used to find the link.
- link callback: In this callback you specify your link options.

Example

This is how you would add a link. .. code:

```
$testMenu->link('home', function (Link $link) {  
    $link->name = 'Home';  
    $link->url = route('home');  
});
```

Adding a drop down

Add a drop down to the menu.

Parameters

- slug: The slug used to find the link.
- text: The text to be displayed for the dropdown
- dropdown callback: In this callback you specify your drop down options.

Example

This is how you would add a drop down. .. code:

```
$testMenu->dropDown('user', 'Admin', function (DropDown $dropDown) {
    $dropDown->link('profile.edit', function (Link $link) {
        $link->name = 'Edit your profile';
        $link->url = 'user/profile/'; // static url
    });
    $dropDown->link('logout', function (Link $link) {
        $link->name = 'Logout';
        $link->url = route('logout'); // Laravel routed url
    });
});
```

1.2.4 Links

Links are single objects under a menu that are meant to be used to create an html link. You have the option to create a link manually or via the quick link method.

```
$menu->link('slug', function (Link $link) {
    $link->name = 'Take me to google';
    $link->url = 'http://google.com';
    $link->options = ['class' => 'btn'];
});
```

Name

The name property is used to create the human readable link text.

Set Name

Set the name of the link

```
$link->name = 'Name';
```

Get Name

The name can be accessed via the name property

```
$link->name
```

URL

The url is the address the link will point to.

Set Url

Set the url of a link

```
$link->url = 'http://nukacode.com';
```

Get Url

The url can be accessed via the url property

```
$link->url
```

Options

Options can be used for extra properties for the link like: class, alt-text, style, exc.

Set Options

Set the options array

Note: Options are saved as an array. Adding options just merges the new options into the array.

```
$link->options = [  
    'class' => 'btn-warning'  
];
```

Get Options

The options can be accessed via the options property

```
$this->options
```

Active Flag

The active property is used to show which link is currently active. (Default: false)

Set Active Flag

Set the active flag. The default parameter for setActive is true

```
// Set link to active  
$link->setActive();  
$link->setActive(true);  
  
// Set link to inactive  
$link->setActive(false);
```

Get Active Flag

The active flag can be accessed via the `isActive()` method.

```
$link->isActive()
```

Insert After/Before

You can tell a link to be inserted before or after another link or dropdown.

Parameters: slug: The slug of the link or drop down to insert before or after.

```
$link->insertAfter('slug');  
$link->insertBefore('slug');
```

1.2.5 Drop Downs

A drop down is just a link that allows you to add sub links.

This Class also implements the linkable trait so it has access to all the link and drop down add methods in the menu class.

Note: See link documentation for methods and properties when adding links.

Parameters: Slug: The slug used to find this drop down again when trying to edit it, when setting the drop down as active or when inserting before or after another menu item.

Link Name: The name of the drop down.

```
$menu->dropDown('slug', 'drop down name', function (DropDown $dropDown) {  
    $dropDown->link('slug', function (Link $link) {  
        $link->name = 'Take me to google';  
        $link->url = 'http://google.com';  
        $link->options = ['class' => 'btn'];  
    });  
  
    $dropDown->link('slug2', function (Link $link) {  
        $link->name = 'Take me to git hub';  
        $link->url = 'http://github.com';  
        $link->options = ['class' => 'btn btn-primary'];  
    });  
});
```

Get drop down

You can use the get dropdown method to pull back the drop down object and change it.

```
$menu->getDropDown('slug', function (DropDown $dropDown) {  
    $dropDown->name = 'Change the dropdown name';  
});
```

Check if the drop down has links

You can call the `hasLinks()` method on the drop down class to see if there are any links. This is to allow you to run a foreach if there are links present. Otherwise treat it as a normal link.

Insert After/Before

You can tell a link to be inserted before or after another link or dropdown.

Parameters: slug: The slug of the link or drop down to insert before or after.

```
$dropdown->insertAfter('slug');  
$dropdown->insertBefore('slug');
```

Determine Active Parentage

By default, a drop down will become active when a child link becomes active. Most of the time this is how it should be, but in case that is not desirable there is a helper method to disable this.

```
$dropdown->disableActiveParentage();
```

1.2.6 Middleware

Middleware is provided with the menu package. This middle ware allows you to set which links are marked as active. This is done via the laravel routes file.

This middleware is compatible with Laravel 5.0 and 5.1.

Laravel 5.0

In laravel 5.0 you need to specify what middle ware you will use and the active link.

```
Route::get('/', [  
    'middleware' => 'menu',  
    'active' => 'home', // The slug of your menu item.  
    'as' => 'home',  
    'uses' => 'HomeController@index'  
]);
```

Laravel 5.1

In laravel 5.1 you can set the middle ware and the active item all in one line.

```
Route::get('/', [  
    'middleware' => 'menu:home', // menu is the middle ware and home is the slug of your menu item.  
    'as' => 'home',  
    'uses' => 'HomeController@index'  
]);
```

Notes

This middle ware can also be used in the `Route::group` method as well.

```
Route::group(['namespace' => 'App\Http\Controllers', 'middleware' => 'menu:home'], function () {
    Route::get('/', [
        'as' => 'home',
        'uses' => 'HomeController@index'
    ]);
});
```

1.2.7 Examples

Full code sample

Here is a full menu example.

```
$leftMenu = \Menu::getMenu('leftMenu');
$rightMenu = \Menu::getMenu('rightMenu');

// Home
$leftMenu->link('home', function (Link $link) {
    $link->name = 'Home';
    $link->url = route('home');
});

if (\Auth::guest()) {
    $rightMenu->link('login', function (Link $link) {
        $link->name = 'Login';
        $link->url = route('login');
    });

    $rightMenu->link('register', function (Link $link) {
        $link->name = 'Register';
        $link->url = route('register');
    });
} else {
    \Auth::user()->updateLastActive();

    if (\Auth::user()->is('ADMIN')) {
        $rightMenu->dropDown('admin', 'Admin', function (DropDown $dropDown) {
            $dropDown->link('dashboard', function (Link $link) {
                $link->name = 'Dashboard';
                $link->url = route('admin.index');
            });
        });
    }

    $rightMenu->dropDown('user', \Auth::user()->username, function (DropDown $dropDown) {
        $dropDown->link('profile.edit', function (Link $link) {
            $link->name = 'Edit your profile';
            $link->url = 'user/profile/';
        });
        $dropDown->link('logout', function (Link $link) {
            $link->name = 'Logout';
            $link->url = route('logout');
        });
    });
}
```

```

});

$rightMenu->link('profile.view', function (Link $link) {
    $link->name = 'Public profile';
    $link->url = 'user/view/' . \Auth::user()->id;
    $link->insertAfter('profile.edit');
});
}

```

Output

```

NukaCode\Menu\Menu Object
(
    [name] => right_menu
    [links] => Illuminate\Support\Collection Object
        (
            [items:protected] => Array
                (
                    [0] => NukaCode\Menu\Link Object
                        (
                            [menu] => NukaCode\Menu\Menu Object
*RECURSION*
                            [slug] => login
                            [name] => Login
                            [url] => http://localhost:5565/login
                            [options] => Array
                                (
                                    )
                            [insert] =>
                            [active] =>
                        )
                    [1] => NukaCode\Menu\Link Object
                        (
                            [menu] => NukaCode\Menu\Menu Object
*RECURSION*
                            [slug] => register
                            [name] => Register
                            [url] => http://localhost:5565/register
                            [options] => Array
                                (
                                    )
                            [insert] =>
                            [active] =>
                        )
                )
            )
        [insert] =>
    )
NukaCode\Menu\Menu Object
(
    [name] => left_menu

```

```
[links] => Illuminate\Support\Collection Object
(
    [items:protected] => Array
        (
            [0] => NukaCode\Menu\Link Object
                (
                    [menu] => NukaCode\Menu\Menu Object
*RECURSION*
                    [slug] => home
                    [name] => Home
                    [url] => http://localhost:5565
                    [options] => Array
                        (
                        )
                    [insert] =>
                    [active] =>
                )
        )
    )
    [insert] =>
)
```