

Installation

Use An Empty Laravel 6.x Starter Project

From the main `.zip` file downloaded from ThemeForest, unpack the `Stack-Laravel-Project.zip` archive.

Then change directory to the project (we'll assume it's called `stack-laravel`) and run:

```
composer install
```

Create A New Laravel 6.x Project

Via Laravel Installer. First, download the Laravel installer using Composer:

```
composer global require laravel/installer
```

Make sure to place Composer's system-wide vendor bin directory in your `$PATH` so the laravel executable can be located by your system. This directory exists in different locations based on your operating system; however, some common locations include:

- macOS and GNU / Linux Distributions: `$HOME/.composer/vendor/bin`
- Windows: `%USERPROFILE%\AppData\Roaming\Composer\vendor\bin`

Then create a new Laravel project:

```
laravel new stack-laravel
```

Setup Local Development Server

Using the built-in development server, you can access the site in your browser at <http://127.0.0.1:8000>:

```
php artisan serve
```

Optionally, setup [Laravel Valet](#) if you're on a Mac:

```
valet link stack-laravel
```

After running the link command, you can access the site in your browser at <http://stack-laravel.test>

Setup Stack Laravel Package

From the main `.zip` file downloaded from ThemeForest, move the `Stack-Laravel-Package.zip` file to your project's `app/` directory and unpack, after which you should have `app/Stack`.

Then register the Stack Service Provider within `app/Providers/AppServiceProvider.php`:

```
<?php

use App\Stack\StackServiceProvider;

class AppServiceProvider extends ServiceProvider
{
    public function register()
    {
        $this->app->register(
            StackServiceProvider::class
        );
    }
}
```

Global helpers

The views are using a few helpers to set page titles and set active classes based on the current route, etc. Update `composer.json` to autoload the helpers file:

```
"autoload": {
    "files": [
        "app/Stack/helpers.php"
    ],
}
```

And then run `composer dump-autoload`.

Assets

To copy the configuration and assets included with the package within your project, run:

```
php artisan vendor:publish --provider="App\Stack\StackServiceProvider" --force
```

The publish command will create the following files:

Laravel Mix Customizations

```
theme-mix.yaml
ThemeMix.js
webpack.mix.js
```

Configuration

```
config/stack.php
```

Blade views, layouts, components

```
resources/views/vendor/stack
```

Source Scss and JavaScript files

```
resources/js
resources/sass
```

Dependencies

Update `package.json` with the following:

```
"dependencies": {
  "stack": "github:themefy/stack#master"
},
"devDependencies": {
  "@babel/plugin-syntax-dynamic-import": "^7.2.0",
  "@babel/polyfill": "^7.2.5",
  "axios": "^0.19",
  "cross-env": "^5.2.0",
  "laravel-mix": "^4.1.2",
  "laravel-mix-merge-manifest": "^0.1.2",
  "lodash": "^4.17.13",
  "resolve-url-loader": "^2.3.1",
  "theme-mix": "^1.2.0"
}
```

Then, to install the dependencies, run:

```
npm install
```

Authentication

Update `.env` database connection:

```
DB_CONNECTION=sqlite
```

Comment out or remove the `.env` database name:

```
# DB_DATABASE=laravel
```

Create a database file:

```
touch database/database.sqlite
```

Run the database migrations:

```
php artisan migrate
```

Create demo credentials:

```
php artisan db:seed --class="App\Stack\UsersTableSeeder"
```

Development

You can run the usual Laravel Mix commands to compile all the assets.

```
npm run watch
npm run development
npm run production
```

BrowserSync

Running `npm run watch` will open your browser with BrowserSync enabled by default on <http://localhost:3000>

For BrowserSync, you may need to update `theme-mix.yaml` with your local project hostname (Vagrant or Valet hostname):

```
proxy: http://stack-laravel.test
```

If using the built-in development server via `php artisan serve`, then the proxy should be `http://127.0.0.1:8000`

You can disable BrowserSync from `webpack.mix.js`:

```
mix.themeMix({
  runTasks: {
    ...
    browserSync: false
  }
})
```

Hot Module Replacement

If using Hot Module Replacement i.e. `npm run hot` you may need to update `webpack.mix.js` with your local project hostname (Vagrant or Valet):

```
mix.options({
  hmrOptions: {
    host: 'stack-laravel.test',
    port: 8080
  }
})
```

If using the built-in development server via `php artisan serve`, then the hostname should be `127.0.0.1`

Routes

By default, all routes are under the `/` prefix, so accessing the login page for example you would go to <http://localhost:3000/login>

You can change that from `config/stack.php`, for example to add a prefix to all routes:

```
'path' => '/stack', // adds a prefix to all routes  
'path' => null,    // removes the prefix
```