

Laravel Modules 8/24/2021

<https://nwidart.com/laravel-modules/v6/introduction>

Prerequisites: It assumes you have [PHP](#), [MySQL](#), [Composer](#), [Node.js](#), [Npm](#) installed on your system.

Run the commands marked in **bold dark-blue**

```
composer create-project --prefer-dist laravel/laravel:^7.0 project_name
composer create-project --prefer-dist laravel/laravel project_name
```

```
composer create-project --prefer-dist laravel/laravel laravel-modules
```

```
cd project_name
php artisan -V
Laravel Framework 8.56.0
```

```
git init
git add .
git commit -m "Comment"
```

```
git remote add origin https://github.com/SergeyHub/repository-name.git
git push -u origin master
```

Edit env. file

```
mysql -u root -p
create database db_name;
drop database db_name;
exit
```

```
composer require laravel/ui
composer require laravel/ui "^2.4" 7
```

```
php artisan ui bootstrap
php artisan ui bootstrap --auth

php artisan ui vue
php artisan ui vue --auth

npm i axios vform
```

```
npm install && npm run dev
npm run watch

php artisan migrate
php artisan serve
```

```
composer require nwidart/laravel-modules

php artisan vendor:publish --
provider="Nwidart\Modules\LaravelModulesServiceProvider"
```

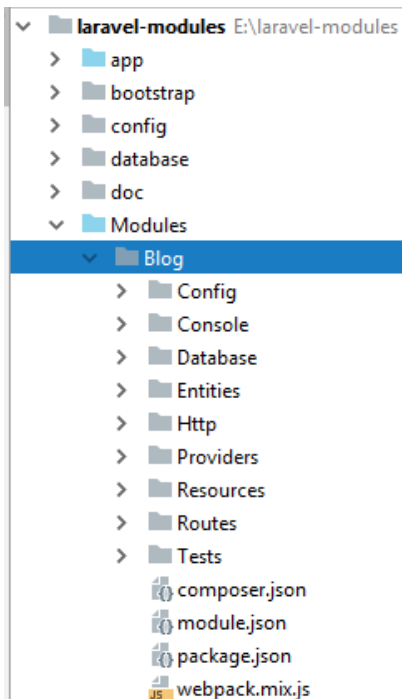
Autoloading

By default the module classes are not loaded automatically. You can autoload your modules using **psr-4**. For example :

```
"autoload": {
    "psr-4": {
        "App\\": "app/",
        "Database\\Factories\\": "database/factories/",
        "Database\\Seeders\\": "database/seaders/",
        "Modules\\": "Modules/"
    }
}
```

Tip: Don't forget to run `composer dump-autoload` afterwards

```
php artisan module:make Blog
```



php artisan list

module	
module:delete	Delete a module from the application
module:disable	Disable the specified module.
module:dump	Dump-autoload the specified module or for all module.
module:enable	Enable the specified module.
module:install	Install the specified module by given package name (vendor/name).
module:list	Show list of all modules.
module:make	Create a new module.
module:make-command	Generate new Artisan command for the specified module.
module:make-controller	Generate new restful controller for the specified module.
module:make-event	Create a new event class for the specified module
module:make-factory	Create a new model factory for the specified module.
module:make-job	Create a new job class for the specified module
module:make-listener	Create a new event listener class for the specified module
module:make-mail	Create a new email class for the specified module
module:make-middleware	Create a new middleware class for the specified module.
module:make-migration	Create a new migration for the specified module.
module:make-model	Create a new model for the specified module.
module:make-notification	Create a new notification class for the specified module.
module:make-policy	Create a new policy class for the specified module.
module:make-provider	Create a new service provider class for the specified module.
module:make-request	Create a new form request class for the specified module.
module:make-resource	Create a new resource class for the specified module.
module:make-rule	Create a new validation rule for the specified module.
module:make-seed	Generate new seeder for the specified module.
module:make-test	Create a new test class for the specified module.
module:migrate	Migrate the migrations from the specified module or from all modules.
module:migrate-refresh	Rollback & re-migrate the modules migrations.
module:migrate-reset	Reset the modules migrations.
module:migrate-rollback	Rollback the modules migrations.
module:migrate-status	Status for all module migrations
module:publish	Publish a module's assets to the application
module:publish-config	Publish a module's config files to the application
module:publish-migration	Publish a module's migrations to the application
module:publish-translation	Publish a module's translations to the application
module:route-provider	Create a new route service provider for the specified module.
module:seed	Run database seeder from the specified module or from all modules.
module:setup	Setting up modules folders for first use.
module:unuse	Forget the used module with module:use
module:update	Update dependencies for the specified module or for all modules.
module:use	Use the specified module.
module:v6:migrate	Migrate laravel-modules v5 modules statuses to v6.

php artisan module:delete Blog

<http://127.0.0.1:8000/blog>

← → ↺

🔍 Найдите в Google или введите адрес

Hello World

This view is loaded from module: Blog

php artisan module:make ShoppingCart Cart News Product Promo

```
e:\laravel-modules>php artisan module:list
```

Name	Status	Priority	Path
Blog	Enabled	0	E:\laravel-modules\Modules\Blog
Cart	Enabled	0	E:\laravel-modules\Modules\Cart
News	Enabled	0	E:\laravel-modules\Modules\News
Product	Enabled	0	E:\laravel-modules\Modules\Product
Promo	Enabled	0	E:\laravel-modules\Modules\Promo
ShoppingCart	Enabled	0	E:\laravel-modules\Modules\ShoppingCart

Edit E:\laravel-modules\resources\views\layouts\app.blade.php

```
<div class="collapse navbar-collapse" id="navbarSupportedContent">
    <!-- Left Side Of Navbar -->
    <ul class="navbar-nav mr-auto">

        @if(Module::collections()->has('Blog'))
            <li class="nav-item">
                <a class="nav-link" href="/blog">Blog</a>
            </li>
        @endif

        @if(Module::collections()->has('Cart'))
            <li class="nav-item">
                <a class="nav-link" href="/cart">Cart</a>
            </li>
        @endif

        @if(Module::collections()->has('News'))
            <li class="nav-item">
                <a class="nav-link" href="/news">News</a>
            </li>
        @endif

        @if(Module::collections()->has('Product'))
            <li class="nav-item">
                <a class="nav-link" href="/product">Product</a>
            </li>
        @endif

        @if(Module::collections()->has('Promo'))
            <li class="nav-item">
                <a class="nav-link" href="/promo">Promo</a>
            </li>
        @endif

    </ul>
```

Edit E:\laravel-modules\Modules\Blog\Resources\views\index.blade.php

```

@extends('layouts.app')

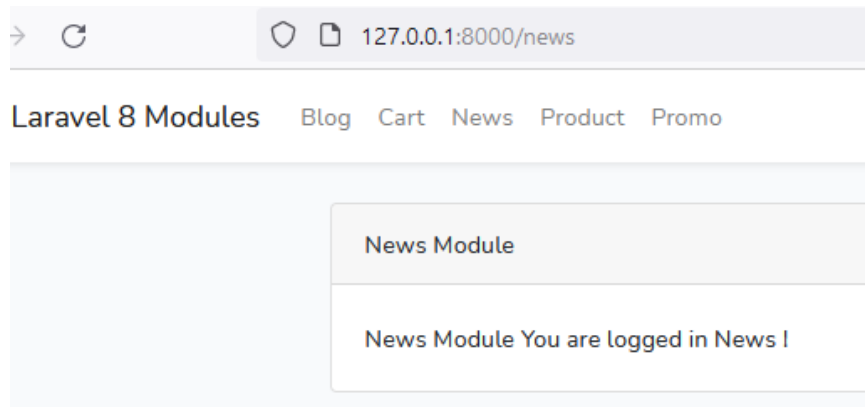
@section('content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-8">
                <div class="card">
                    <div class="card-header">{{ __('Blog Module') }}</div>

                    <div class="card-body">
                        @if (session('status'))
                            <div class="alert alert-success" role="alert">
                                {{ session('status') }}
                            </div>
                        @endif
                        Blog Module
                        {{ __('You are logged in Blog !') }}
                    </div>
                </div>
            </div>
        </div>
    </div>
@endsection

@section('content')
    <h1>Hello World</h1>

    <p>
        This view is loaded from module: {!! config('blog.name') !!}
    </p>
@endsection

```



```

sudo chmod -R 775 storage/framework
sudo chmod -R 775 ./storage/logs
chmod -R 775 bootstrap/cache

sudo service apache2 restart

```

sudo chmod -R 777 /var/www/html Never use 777. With 777 permissions, you are giving anyone with a connection full access to the files or directories with those permissions. They may alter them in any way they choose, including maliciously. Many account hacking incidents stem from 777 permissions.

the requested url was not found on this server. Laravel

1. edit the apache2 conf to allow for symlinks etc
`sudo nano /etc/apache2/apache2.conf`

2. Add this to the file

```
<Directory /var/www/html/>  
  Options Indexes FollowSymLinks  
  AllowOverride all  
  Require all granted  
</Directory>
```

3. Restart your apache2 server

```
sudo service apache2 restart
```