Steps to integrate Scheduler with existing Laravel application.

1. Create modules folder if doesn’t exist, place Scheduler folder under modules folder.
2. Put below code in composer.json

"require": {

"php": ">=7.0.0",

"fideloper/proxy": "~3.3",

"guzzlehttp/guzzle": "^6.3",

"laravel/framework": "5.5.\*",

"swiftmailer/swiftmailer": "@stable",

"laravel/tinker": "~1.0"

},

"autoload": {

"classmap": [

"database/seeds",

"database/factories"

],

"psr-4": {

"App\\": "app/",

"Modules\\Scheduler\\": "modules/Scheduler/src/"

}

},

1. Add following line to .env file. This setting is useful for sending mail using scheduler.

MAIL\_DRIVER=smtp

MAIL\_HOST=smtp.gmail.com

MAIL\_PORT=587

MAIL\_USERNAME=testmili@gmail.com

MAIL\_PASSWORD=kembtbgueyxbpobs

MAIL\_ENCRYPTION=tls

1. Add Modules\Scheduler\SchedulerServiceProvider::class, in your config/app.php file in the providers array.
2. Run  composer dump-autoload –o
3. Execute php artisan vendor:publish --provider="Modules\Scheduler\ SchedulerServiceProvider"
4. Run migration for scheduler table = php artisan migrate
5. Create event and their listeners by using following link.
6. If you find any difficulty while creating events please follow this link

<http://itsolutionstuff.com/post/how-to-create-event-for-mail-sending-in-laravel-52example.html>

1. Then copy files from Rootfolder\app\Events\ in Event folder.
2. Then copy files from Rootfolder\app\Listeners\ in Listener folder.
3. Copy Rootfolder\resources\views\home.blade.php to views folder.
4. Now create artisan console command class by using following command.

php artisan make:console DailyScheduler

1. Now replace Rootfolder\app\Console\Commands\DailyScheduler.php in \app\Console\Commands folder.
2. Now go to working folder. – cd wamp64\www\Rootfolder.
3. Now run scheduler manually by using following command.

php artisan DailyScheduler:dailyschedule

1. We can run all scheduler manually by using below command.

php artisan schedule:run