

It's teamwork, but simpler, more pleasant and more productive.

AppDividend

magicbricks
iFollow



2, 3 BHK Flat
Green Island
Rajarhat, Kolkata

Home > Laravel >



LARAVEL

Laravel 5.5 Artisan Console Tutorial

Laravel Command Tutorial



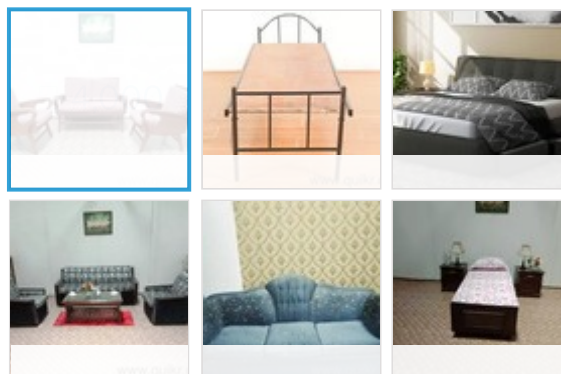
By Krunal

Last updated Feb 22, 2018



By clicking the subscribe button you will never miss the new articles!

 **Subscribe**



Laravel 5.5 Artisan Console Tutorial is today's topic. Artisan Command Line Interface comes with It's teamwork, but simpler, more pleasant and more productive. Brand new Laravel project out of the box. There are so many commands Laravel application provides us by default. There so many tasks in your application that can be very well handled by command line tools. So this tutorial is dedicated to Laravel 5.5 Commands. You can list all the Laravel 5.5 Commands by the typing following command.

```
php artisan list
```

Content Overview [hide]

- 1 Laravel 5.5 Artisan Console Tutorial
- 2 Step 1: Make one Laravel 5.5 Command
- 3 Step 2: Register the command in Laravel 5.5.
- 4 Step 3: Structure of this SharesAlloted.php file.
- 5 Step 5: Command Structure
 - 5.0.1 Arguments
 - 5.0.2 Options
- 6 Step 6: Write the logic for this command.

Laravel 5.5 Artisan Console Tutorial

If you are new to Laravel 5.5 then check out my these tutorials on this blog.

1. **Laravel 5.5 Tutorial With Example From Scratch**
2. **Laravel 5 Middleware Tutorial With An Example**

Step 1: Make one Laravel 5.5 Command

We can also build our own commands that suit our application needs. For that, we need to type the following command.

```
php artisan make:command SharesAlloted
```

This command will generate one directory inside **console** directory called **Commands**. In that directory, we can find the file **SharesAlloted.php**

So command file is in the **Console >> Commands >> SharesAlloted.php**

Step 2: Register the command in Laravel 5.5.

We need to add it into our `app/console/Kernel.php` file in the `$commands` array.

<?php It's teamwork, but simpler, more pleasant and more productive. X

```
// Kernel.php

/**
 * The Artisan commands provided by your application.
 *
 * @var array
 */
protected $commands = [
    'App\Console\Commands\SharesAlloted'
];
```

So, now we have registered the command to the Laravel Application.

Step 3: Structure of this SharesAlloted.php file.

<? h, It's teamwork, but simpler, more pleasant and more productive. X

```
// SharesAlloted.php

namespace App\Console\Commands;

use Illuminate\Console\Command;

class SharesAlloted extends Command
{
    /**
     * The name and signature of the console command.
     *
     * @var string
     */
    protected $signature = 'command:name';

    /**
     * The console command description.
     *
     * @var string
     */
    protected $description = 'Command description';

    /**
     * Create a new command instance.
     *
     * @return void
     */
    public function __construct()
    {
        parent::__construct();
    }

    /**
     * Execute the console command.
     *
     * @return mixed
     */
    public function handle()
    {
        //
    }
}
```

Next, we should fill the two properties of this class.

1. **\$signature**
2. **\$decription**

```

// Step 5: Alloted.php, but simpler, more pleasant and more productive.

/**
 * The name and signature of the console command.
 *
 * @var string
 */
protected $signature = 'alloted:shares {user}';

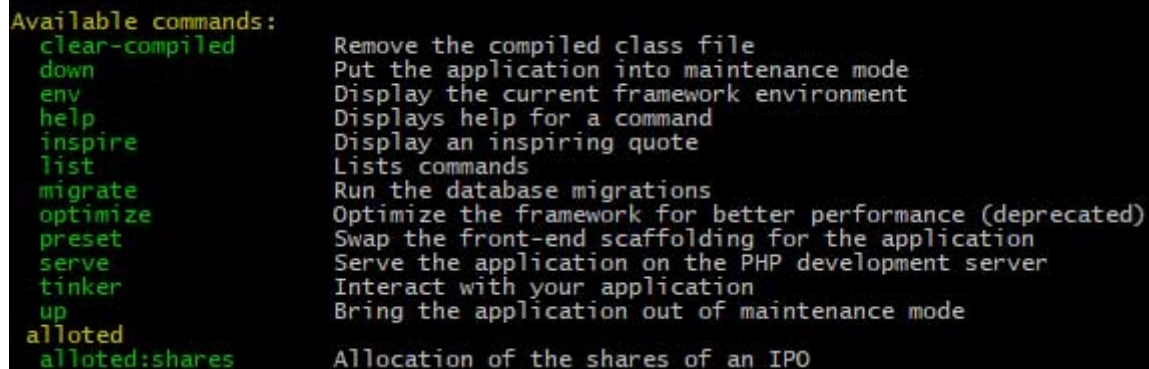
/**
 * The console command description.
 *
 * @var string
 */
protected $description = 'Allocation of the shares of an IPO';

```

Now, go to your terminal and type the following command.

```
php artisan list
```

You can see our new registered command.



```

Available commands:
clear-compiled  Remove the compiled class file
down           Put the application into maintenance mode
env           Display the current framework environment
help         Displays help for a command
inspire      Display an inspiring quote
list         Lists commands
migrate       Run the database migrations
optimize      Optimize the framework for better performance (deprecated)
preset       Swap the front-end scaffolding for the application
serve        Serve the application on the PHP development server
tinker       Interact with your application
up           Bring the application out of maintenance mode
alloted
alloted:shares  Allocation of the shares of an IPO

```

Step 5: Command Structure

It's teamwork, but simpler, more pleasant and more productive.



×

```
/**
 * The name and signature of the console command.
 *
 * @var string
 */
protected $signature = 'alloted:shares {user} {age} {--difficulty=} {--istest=}';
```

{user} {--difficulty=} {--istest=} these are optional arguments.

Arguments

All user-supplied arguments and options are wrapped in curly braces.

```
/**
 * The name and signature of the console command.
 *
 * @var string
 */
protected $signature = 'alloted:shares {user}';
```

In above example {user} is argument.

We can also make an argument optional.

```
protected $signature = 'alloted:shares {user?}';
```

Options

Options, like arguments, are another form of user input.

Options are prefixed by two hyphens (--) when they are specified on the command line.

```

/**
 * It's teamwork, but simpler, more pleasant and more productive.
 * The name and signature of the console command.
 *
 * @var string
 */
protected $signature = 'alloted:shares {user} [--listing]';

```

If you need to retrieve all of the arguments as an array call the `arguments` method

```
$arguments = $this->arguments();
```

If you want to retrieve specific options then please use the following command.

```

// Retrieve a specific option...
$queueName = $this->option('listing');

// Retrieve all options...
$options = $this->options();

```

Step 6: Write the logic for this command.

In the **SharesAlloted.php** file, there is the function called **handle**. This function contains all the logic of the command.

We are just displaying a single line of this command. So we just write two lines of code in the **handle()** method.

```

/**
 * Execute the console command.
 *
 * @return mixed
 */
public function handle()
{
    $user = $this->argument('user');
    $this->line('Hello ' . $this->argument('user'));
}

```

Switch to the terminal and type the following command.

```
php artisan alloted:shares AppDividend
```

So this will print the following line in the terminal.

Hello AppDividend

The whole file looks like this.

It's teamwork, but simpler, more pleasant and more productive.



```
<?php

// SharesAlloted.php

namespace App\Console\Commands;

use Illuminate\Console\Command;

class SharesAlloted extends Command
{
    /**
     * The name and signature of the console command.
     *
     * @var string
     */
    protected $signature = 'alloted:shares {user}';

    /**
     * The console command description.
     *
     * @var string
     */
    protected $description = 'Allocation of the shares of an IPO';

    /**
     * Create a new command instance.
     *
     * @return void
     */
    public function __construct()
    {
        parent::__construct();
    }

    /**
     * Execute the console command.
     *
     * @return mixed
     */
    public function handle()
    {
        $user = $this->argument('user');
        $this->line('Hello ' . $this->argument('user'));
    }
}
```

So this is the basic application of **Commands in Laravel 5.5**

You can find the full documentation in the following URL.

★ <https://laravel.com/docs/5.5/artisan>
It's teamwork, but simpler, more pleasant and more productive.

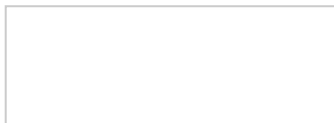


If you have doubt in this **Laravel 5.5 Artisan Console Tutorial** then ask in a comment.

[Artisan Command](#)[Laravel 5.5](#)[Laravel 5.5 Tutorial](#)

Krunal



I am Web Developer and Blogger. I have created this website for the web developers to understand complex concepts in an easy manner.




Plot in Action Area 3, Kolkata

7.8 Lac

[Enquire Now!](#)



 [report this ad](#)

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

