ARYATECHNO

Laravel - Artisan Console

Topics : Laravel
Written on December 25, 2023

Artisan is the command-line interface included with Laravel. It provides a number of helpful commands for common tasks like database migrations, seeding, testing, and more.

1. Running Artisan Commands:

To run an Artisan command, use the `php artisan` command followed by the desired command.

```
php artisan list
```

This will display a list of all available Artisan commands.

2. Creating a Custom Artisan Command:

You can create your own Artisan commands using the `make:command` Artisan command.

```
php artisan make:command MyCustomCommand
```

This will generate a new command class in the `app/Console/Commands` directory. You can then define the behavior of your command in the generated class.

3. Running a Custom Artisan Command:

After creating a custom command, you can run it using the `php artisan` command.

```
php artisan my:custom-command
```

4. Command Signature and Description:

You can define the signature and description of your custom command in the generated class.

```
/**
 * The name and signature of the console command.
 *
 * @var string
 */
protected $signature = 'my:custom-command';
```

```
/**
 * The console command description.
 *
 * @var string
 */
protected $description = 'My custom Artisan command description.';
```

5. Command Options and Arguments:

You can define options and arguments for your custom command.

```
/**
 * The console command signature with options and arguments.
 *
 * @var string
 */
protected $signature = 'my:custom-command {--option} {argument}';
```

6. Running Migrations:

Artisan includes commands for running migrations.

```
php artisan migrate
```

7. Rolling Back Migrations:

To roll back the last database migration, you can use the `migrate:rollback` command.

```
php artisan migrate:rollback
```

8. Seeding the Database:

Artisan provides commands for seeding the database.

```
php artisan db:seed
```

9. Creating a Controller:

You can create controllers using the `make:controller` command.

```
php artisan make:controller MyController
```

10. Creating a Model:

To create a model, you can use the `make:model` command.

```
php artisan make:model MyModel
```

11. Creating a Migration:

To generate a new database migration, you can use the `make:migration` command.

```
php artisan make:migration create_table_name
```

12. Viewing Routes:

To see a list of all registered routes, you can use the `route:list` command.

```
php artisan route:list
```

13. Creating a Job:

You can generate a new job using the `make:job` command.

```
php artisan make:job MyJob
```

14. Optimizing for Production:

To optimize the application for production, you can use the `optimize` command.

```
php artisan optimize
```

15. Artisan Tinker:

Artisan Tinker is a REPL (Read-Eval-Print Loop) that allows you to interact with your application from the command line.

```
php artisan tinker
```