

La Commande “php artisan”



La commande « `php artisan` » fait partie du framework Laravel, un framework d’application Web PHP populaire. « Artisan » est l’interface de ligne de commande incluse avec Laravel et fournit plusieurs commandes utiles pour développer votre application. Pour utiliser `php artisan`, vous devez avoir Laravel installé sur votre système.

Voici quelques commandes « `php artisan` » courantes et leurs utilisations :

1. Commandes d’application :

- `php artisan serve` : Démarre un serveur de développement à `http://localhost:8000`.
- `php artisan tinker` : Ouvre un shell interactif pour interagir avec votre application.

2. Commandes de base de données :

- `php artisan migrate` : Exécute les migrations de bases de données.
- `php artisan migrate:rollback` : annule la dernière migration de la base de données.
- `php artisan db:seed` : amorce la base de données avec des enregistrements.

3. Commandes de création :

- `php artisan make:controller ControllerName` : Crée un nouveau contrôleur.
- `php artisan make:model ModelName` : Crée un nouveau modèle.

- `php artisan make:migration create_table_name_table` : Crée un nouveau fichier de migration.

4. Commandes de cache :

- `php artisan cache:clear` : Efface le cache de l'application.
- `php artisan config:cache` : Crée un fichier cache pour un chargement plus rapide de la configuration.

5. Commandes d'itinéraire :

- `php artisan route:list` : Répertoire toutes les routes enregistrées.

6. Commandes de file d'attente :

- `php artisan queue:work` : démarre le traitement des tâches dans la file d'attente.
- `php artisan queue:listen` : écoute une file d'attente et traite les tâches au fur et à mesure qu'elles sont placées dans la file d'attente.

7. Commandes de test :

- `php artisan test` : Exécute les tests de l'application.

Exemple d'utilisation de base

Command	Action
<code>php artisan serve</code>	To run laravel application
<code>php artisan make:Controller ControllerName</code>	To make new controller
<code>php artisan make:Model ModelName</code>	To make new model
<code>php artisan make:Middleware MiddlewareName</code>	To make middleware class
<code>php artisan make:migration tablename</code>	To make migration file
<code>php artisan migrate</code>	To run migration
<code>php artisan cache:clear</code>	Clear Application Cache
<code>php artisan route:clear</code>	Clear Route Cache
<code>php artisan config:clear</code>	Clear Config Cache
<code>php artisan view:clear</code>	Clear Compiled View Files

Pour utiliser la commande « `php artisan` », accédez au répertoire racine de votre projet Laravel dans votre terminal ou invite de commande et entrez la commande souhaitée. Par exemple, pour démarrer le serveur de développement, vous exécuteriez :

```
php artisan serve
```

Création d'un nouveau projet Laravel

```
➔ code composer create-project laravel/laravel test dev-develop
Installing laravel/laravel (dev-develop d6cad21cb2288713d9c09a31f9b4db86f116039)
- Installing laravel/laravel (dev-develop develop): Cloning develop from cache
Created project in test
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 71 installs, 0 updates, 0 removals
- Installing vlucas/phpdotenv (v2.5.1): Loading from cache
- Installing symfony/css-selector (v4.1.3): Loading from cache
- Installing tijsverkoyen/css-to-inline-styles (2.2.1): Loading from cache
- Installing symfony/polyfill-php72 (v1.9.0): Loading from cache
- Installing symfony/polyfill-mbstring (v1.9.0): Loading from cache
- Installing symfony/var-dumper (v4.1.3): Loading from cache
- Installing symfony/routing (v4.1.3): Loading from cache
- Installing symfony/process (v4.1.3): Loading from cache
- Installing symfony/polyfill-ctype (v1.9.0): Loading from cache
- Installing symfony/http-foundation (v4.1.3): Loading from cache
- Installing symfony/event-dispatcher (v4.1.3): Loading from cache
- Installing psr/log (1.0.2): Loading from cache
- Installing symfony/debug (v4.1.3): Loading from cache
- Installing symfony/http-kernel (v4.1.3): Loading from cache
- Installing paragonie/random_compat (v9.99.99): Loading from cache
```

Si vous n'avez pas encore créé de projet Laravel, vous pouvez le faire en utilisant Composer :

```
composer create-project --prefer-dist
laravel/laravel nom-projet
```

Remplacez nom-projet par le nom souhaité pour votre projet.

Commande d'aide

```
$ php artisan list
laravel/framework v6.1.0

Usage:
  command [options] [arguments]

Options:
  -h, --help            Display help for the given command. When no command is given display help for the list command
  -q, --quiet            Do not output any message
  -V, --version          Display this application version
      --ansi|--no-ansi  Force (or disable --no-ansi) ANSI output
  -n, --no-interaction  Do not ask any interactive question
      --env[=ENV]       The environment the command should run under
  -vv|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  clear-compiled  Remove the compiled class file
  down            Start a new database CLI session
  env            Put the application into maintenance / demo mode
  up             Display the current framework environment
  help           Display help for a command
  inspire        Display an inspiring quote
  list           List commands
  migrate        Run the database migrations
  optimize       Cache the framework bootstrap files
  serve          Serve the application on the PHP development server
  test          Run the application tests
  tinker         Interact with your application
  up            Swap the front-end scaffolding for the application
  ui            Bring the application out of maintenance mode
  auth:clear-sets Flush expired password reset tokens
  cache          Flush the application cache
  cache:clear    Flush the application cache
  cache:forget   Remove an item from the cache
  cache:table    Create a migration for the cache database table
  config         Create a cache file for faster configuration loading
  config:cache   Remove the configuration cache file
  config:clear   Remove the configuration cache file
  db:seed        Seed the database with records
  db:seed:drop  Drop all tables, views, and types
  event          Discover and cache the application's events and listeners
  event:cache    Discover and cache the application's events and listeners
  event:clear    Clear all cached events and listeners
  event:generate Generate the missing events and listeners based on registration
  event:list     List the application's events and listeners
  jet           Set the JetAuth secret key used to sign the tokens
  key:secret     Set the JetAuth secret key used to sign the tokens
  key:generate   Set the application key
  log            This is a short description for our newly created command.
  log:write
```

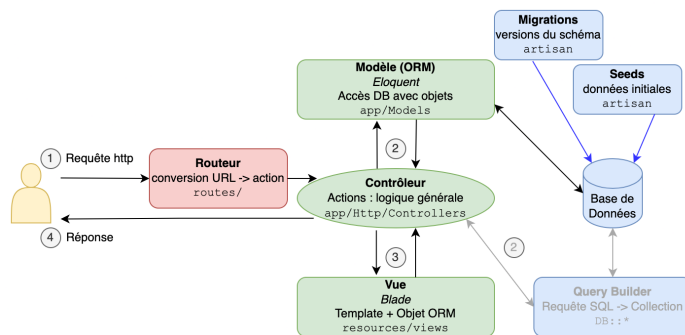
Si vous avez besoin de voir une liste de toutes les commandes « artisan » disponibles, vous pouvez utiliser :

```
php artisan list
```

Et pour obtenir de l'aide sur une commande spécifique :

```
php artisan help nom-commande
```

Remplacez `command-name` par la commande artisan spécifique pour laquelle vous souhaitez obtenir de l'aide.



Conclusion

L'utilisation efficace de « php artisan » peut grandement rationaliser votre flux de travail de développement Laravel.