# Laravel Artisan Laravel CLI

**START HERE**

**`php artisan list`**
See all commands

**`php artisan serve`**
Run the dev server

**`php artisan make:model <Name> -mfc`**
Model, migration, factory, controller

**`php artisan migrate`**
Apply migrations

**`php artisan route:list`**
Show all routes

**`php artisan tinker`**
Open the REPL

**`php artisan test`**
Run the tests

**`php artisan optimize`**
Cache for production

**Route** routes/web.php → *dispatches* → **Controller** make:controller → *uses* → **Model** make:model → *stored by* → **Migration** migrate

## CONCEPTS

Every command has its own help: php artisan help <command>. Add -v to see more detail on failures.

## RUN & INSPECT

**`php artisan list`**
List all commands

**`php artisan help <command>`**
Show a command's options

**`php artisan serve`**
Start the PHP dev server

**`php artisan serve --host=<ip> --port=<n>`**
Serve on another host or port

**`php artisan dev`**
Server, queue, logs and Vite together
*Needs Node 22.13 or later; new in recent Laravel versions.*

**`php artisan tinker`**
Open an interactive REPL

**`php artisan about`**
Overview of app and environment

**`php artisan config:show <file>`**
Print one config file's values

**`php artisan model:show <Model>`**
Show a model's columns and relations

## GENERATE CODE

**`php artisan make:model <Name>`**
Create a model

**`php artisan make:model <Name> -m`**
Model plus migration

**`php artisan make:model <Name> -mfsc`**
Model with migration, factory, seeder, controller

**`php artisan make:model <Name> -a`**
Model plus every companion class

**`php artisan make:controller <Name>`**
Create a controller

**`php artisan make:controller <Name> --resource`**
Controller with CRUD methods

**`php artisan make:migration <name>`**
Create a migration

**`php artisan make:middleware <Name>`**
Create middleware

**`php artisan make:job <Name>`**
Create a queued job

**`php artisan make:test <Name>`**
Create a feature test

## GENERATE CODE (cont.)

**`php artisan make:test <Name> --unit`**
Create a unit test

## DATABASE

**`php artisan migrate`**
Run pending migrations

**`php artisan migrate:status`**
Show which migrations ran

**`php artisan migrate:rollback`**
Undo the last batch

**`php artisan migrate:rollback --step=<n>`**
Undo the last n migrations

**`php artisan migrate:fresh`**
Drop all tables and re-migrate
*Drops every table, including ones not from migrations.*

**`php artisan migrate:fresh --seed`**
Rebuild and seed in one go

**`php artisan migrate --pretend`**
Show the SQL without running it

**`php artisan migrate --force`**
Run in production without a prompt

## DATABASE (cont.)

**`php artisan db:seed`**
Run the DatabaseSeeder

**`php artisan db:seed --class=<Seeder>`**
Run one seeder

**`php artisan schema:dump --prune`**
Squash migrations into a schema file

## ROUTES & CONFIG

**`php artisan route:list`**
List all routes

**`php artisan route:list --path=<prefix>`**
Filter routes by path

**`php artisan route:list --except-vendor`**
Hide routes from packages

**`php artisan route:list -v`**
Show route middleware

**`php artisan install:api`**
Add API routes and Sanctum

**`php artisan config:publish <name>`**
Publish a config file

**`php artisan key:generate`**
Generate the app key

**`php artisan storage:link`**
Link public/storage to storage/app/public

## CACHE & DEPLOY

**`php artisan optimize`**
Cache config, events, routes, views

**`php artisan optimize:clear`**
Remove those caches and the app cache

**`php artisan config:cache`**
Cache the merged config
*After this .env is not read; call env() only inside config files.*

**`php artisan config:clear`**
Drop the cached config

**`php artisan route:cache`**
Cache route registration

**`php artisan view:cache`**
Precompile Blade views

**`php artisan down`**
Enter maintenance mode

**`php artisan up`**
Leave maintenance mode

## QUEUES & SCHEDULE

**`php artisan queue:work`**
Process queued jobs

**`php artisan queue:work --queue=<a>,<b>`**
Process queues in priority order

**`php artisan queue:work --tries=<n>`**
Max attempts per job

## QUEUES & SCHEDULE (cont.)

**`php artisan queue:restart`**
Tell workers to exit after their job

**`php artisan queue:failed`**
List failed jobs

**`php artisan queue:retry all`**
Retry every failed job

**`php artisan schedule:list`**
List scheduled tasks

**`php artisan schedule:work`**
Run the scheduler locally

**`php artisan schedule:run`**
Run due tasks (call from cron)

## TESTING

**`php artisan test`**
Run the test suite

**`php artisan test --filter=<name>`**
Run matching tests only

**`php artisan test --parallel`**
Run tests in several processes
*Needs brianium/paratest as a dev dependency.*

**`php artisan test --coverage`**
Report coverage (Xdebug or PCOV)