



PHP Artisan Tool

The Artisan Console is a great tool that we use for developing our apps. When I mentioned earlier that we generate our Middleware, Controllers, Requests and Models we do so through this file.

If we run `php artisan` you will see a list of commands available that we can run to help with our applications.

Laravel Framework version 5.2.39

Usage:

```
command [options] [arguments]
```

Options:

```
-h, --help           Display this help message
-q, --quiet          Do not output any message
-V, --version        Display this application version
    --ansi           Force ANSI output
    --no-ansi        Disable ANSI output
-n, --no-interaction Do not ask any interactive question
    --env[=ENV]      The environment the command should run under.
-v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose
```

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
list           Lists commands
migrate        Run the database migrations
optimize       Optimize the framework for better performance
serve          Serve the application on the PHP development server
tinker         Interact with your application
up            Bring the application out of maintenance mode
app
  app:name     Set the application namespace
auth
  auth:clear-resets Flush expired password reset tokens
```

cache	
cache:clear	Flush the application cache
cache:table	Create a migration for the cache database table
config	
config:cache	Create a cache file for faster configuration loading
config:clear	Remove the configuration cache file
db	
db:seed	Seed the database with records
event	
event:generate	Generate the missing events and listeners based on registration
key	
key:generate	Set the application key
make	
make:auth	Scaffold basic login and registration views and routes
make:console	Create a new Artisan command
make:controller	Create a new controller class
make:event	Create a new event class
make:job	Create a new job class
make:listener	Create a new event listener class
make:middleware	Create a new middleware class
make:migration	Create a new migration file
make:model	Create a new Eloquent model class
make:policy	Create a new policy class
make:provider	Create a new service provider class
make:request	Create a new form request class
make:seeder	Create a new seeder class
make:test	Create a new test class
migrate	
migrate:install	Create the migration repository
migrate:refresh	Reset and re-run all migrations
migrate:reset	Rollback all database migrations
migrate:rollback	Rollback the last database migration
migrate:status	Show the status of each migration
queue	
queue:failed	List all of the failed queue jobs
queue:failed-table	Create a migration for the failed queue jobs database table
queue:flush	Flush all of the failed queue jobs
queue:forget	Delete a failed queue job
queue:listen	Listen to a given queue
queue:restart	Restart queue worker daemons after their current job
queue:retry	Retry a failed queue job
queue:table	Create a migration for the queue jobs database table
queue:work	Process the next job on a queue
route	
route:cache	Create a route cache file for faster route registration
route:clear	Remove the route cache file
route:list	List all registered routes

schedule	
schedule:run	Run the scheduled commands
session	
session:table	Create a migration for the session database table
spark	
spark:install	Install the Spark scaffolding into the application
spark:kpi	Store the performance indicators for the application
spark:update	Update the Spark installation
spark:update-views	Update the Spark views
spark:version	View the current Spark version
vendor	
vendor:publish	Publish any publishable assets from vendor packages
view	
view:clear	Clear all compiled view files

Generating files

A few commands will be using quite a bit for generating files will be:

`php artisan make:controller OurControllerName` - create a controller file.

`php artisan make:model OurModelName` - create a model file.

`php artisan make:migration MigrationFileName` - create migration file.

`php artisan make:middleware MiddlewareName` - create a middleware file for editing.

`php artisan make:request OurRequestName` - create a request file.

Migration commands

`php artisan migrate:install` - install migration table.

`php artisan migrate` - run our migrations.

`php artisan migrate:refresh` - reset and rerun all database migrations. Think of it like delete all and make them again.

Checking our routes

`php artisan route:list` - checks our list of available routes for the application.