LARAVEL COMMANDS CHEATSHEET

enquiries@welcm.uk | https://welcm.uk | +44 (0)1252 950 650

COMMAND	DESCRIPTION
laravel new project-name	This will create a folder within the folder you are currently in named “project-name” and create all the necessary Laravel project files here. Depending on how your environment is set up, you should be able to navigate to http://project-name.test in your browser and see the default Laravel welcome page.
composer require vendor/package	This will update your composer.json with the necessary details of the package you are choosing to install and then install the package in your project.
composer update	This will look for the newest versions of the packages you have installed and update them. You can manually update the Laravel version in your composer.json file and run this command to update your Laravel version, just make sure there aren’t any breaking changes between versions that need to be addressed.
composer dump-autoload	This updates your vendor/composer/autoload_classmap.php file, you may need to run it if you have a new class in your project that has not yet been loaded.
php artisan list	Lists all the artisan commands, run it and have a read!
php artisan --help OR -h	Displays some basic help, add --help or -h after any of these commands to see the help text with all available flags and options.
php artisan key:generate	This generates a new key and adds it to your .env file. A key is automatically generated when you run laravel new project-name but the command can be useful when cloning an existing project. This app key is mainly used for encrypting cookies.
php artisan --version OR -V	Displays your current version of Laravel
php artisan down	Puts your application into maintenance mode — visitors to the site will see a maintenance message.
php artisan up	Brings your application back out of maintenance mode.
php artisan env	Displays the current environment for your application.
php artisan route:list	Lists all the routes registered in your application.
php artisan serve --host=192.168.1.100 --port=80	Runs a web server that will be accessible locally, if you do not specify a --host or a --port the site will be accessed at your local ip on port 8000.
php artisan make:auth	Creates all that is necessary for authentication in your application. Make sure you run php artisan migrate after this command(see below), then you can navigate to /register or /login on your project to create and log in to an account.
php artisan make:model ModelName -mcr	Creates a model class and file in your project. You can use some, all or none of the -mcr flags when creating a new model -m creates a migration, -c creates a controller and -r specifies that the controller is a resource controller. Run php artisan make:model -h to see the full set of options.

php artisan make:controller ControllerName	Creates a controller file in your project.
php artisan make:migration --table='table' 'description_of_migration'	Creates a database migration file that you can edit to add necessary table properties.
php artisan migrate	Runs any pending database migrations.
php artisan migrate:rollback	Rolls back the latest database migration (ensuring you have the necessary commands in your down() function of the migration).
php artisan migrate:rollback --step=5	This example will roll back the last 5 migrations.
php artisan migrate:reset	Rolls back all migrations.
php artisan vendor:publish	Displays a list of vendor packages installed in your project, giving you the option to specify which you would like to copy the configuration or view files to your own project's folders for additional configuration or customisation.
php artisan config:cache	Speed up your application for production by combining all your config options into a single file that loads quickly.
php artisan route:cache	Speed up your application for production caching all your application's routes.
php artisan route:clear	Clear the cached version of your routes — use this on local deployments if you have cached routes. Re-run the cache command above on production to clear and re-cache routes.
php artisan config:clear	Clear your cached config — use this on local deployments if you have cached config. Re-run the cache command above on production to clear and re-cache config.

Welcm
Software