**Laravel php artisan command**

🧱 **Project Setup / Environment**

| **Command** | **Description** |
| --- | --- |
| php artisan serve | Starts the local development server at http://127.0.0.1:8000 |
| php artisan key:generate | Generates a new APP\_KEY in .env file for encryption |
| php artisan config:cache | Combines all config files into one cached file for faster loading |
| php artisan config:clear | Clears cached configuration |
| php artisan cache:clear | Clears the application cache |
| php artisan route:clear | Clears the cached routes |
| php artisan view:clear | Clears all compiled Blade views |
| php artisan optimize | Optimizes the framework for better performance |
| php artisan optimize:clear | Clears all cache (config, route, view, event) at once |
| php artisan storage:link | Creates a symbolic link between public/storage and storage/app/public |
| php artisan down | Puts the application into maintenance mode |
| php artisan up | Brings the application out of maintenance mode |

**Database & Migrations**

| **Command** | **Description** |
| --- | --- |
| php artisan migrate | Runs all pending migrations to update the database schema |
| php artisan migrate:rollback | Rolls back the last batch of migrations |
| php artisan migrate:reset | Rolls back **all** migrations |
| php artisan migrate:refresh | Rolls back all migrations and re-runs them |
| php artisan migrate:fresh | Drops all tables and re-runs migrations from scratch |
| php artisan make:migration create\_table\_name\_table | Creates a new migration file |
| php artisan db:seed | Runs all seeder classes to populate data |
| php artisan db:seed --class=UserSeeder | Runs a specific seeder class |
| php artisan migrate --seed | Runs migrations and then seeds data |
| php artisan tinker | Opens Laravel’s interactive shell to run PHP code or Eloquent queries directly |

⚙️ **Model, Controller, and Resource Generation**

| **Command** | **Description** |
| --- | --- |
| php artisan make:model Student | Creates a new Eloquent model |
| php artisan make:model Student -m | Creates model with a migration file |
| php artisan make:model Student -mc | Creates model with migration and controller |
| php artisan make:controller StudentController | Creates a controller |
| php artisan make:controller StudentController --resource | Creates a RESTful resource controller (with CRUD methods) |
| php artisan make:controller StudentController --resource --model=Student | Creates a RESTful resource controller (with CRUD methods) for model |
| php artisan make:controller Api/StudentController --api | Creates an API controller (no create/edit views) |
| php artisan make:request StoreStudentRequest | Creates a form request class for validation |
| php artisan make:resource StudentResource | Creates an API resource for transforming JSON output |
| php artisan make:policy StudentPolicy | Creates a policy class for authorization |

💾 **Database Seeding & Factories**

| **Command** | **Description** |
| --- | --- |
| php artisan make:seeder StudentSeeder | Creates a new seeder class |
| php artisan make:factory StudentFactory | Creates a factory for generating fake data |
| php artisan db:seed | Runs the seeder(s) to populate tables |
| php artisan migrate:fresh --seed | Rebuilds DB and seeds it instantly |

🧩 **Routes & Middleware**

| **Command** | **Description** |
| --- | --- |
| php artisan route:list | Displays all registered routes |
| php artisan make:middleware CheckAdmin | Creates a new middleware class |
| php artisan make:provider EventServiceProvider | Creates a new service provider |
| php artisan make:event UserRegistered | Creates an event class |
| php artisan make:listener SendWelcomeEmail | Creates an event listener |

🧑‍💻 **Auth & User Management (Laravel Breeze / Jetstream / UI)**

| **Command** | **Description** |
| --- | --- |
| php artisan make:auth | (Laravel <6) Generates auth scaffolding |
| php artisan ui bootstrap --auth | (Laravel UI) Generates Bootstrap-based auth scaffolding |
| php artisan breeze:install | Installs Laravel Breeze starter kit |
| php artisan jetstream:install livewire | Installs Jetstream with Livewire support |

🔍 Debugging / Information

|  |  |
| --- | --- |
| php artisan list | Lists all available Artisan commands |
| php artisan help migrate | Shows help for a specific command |
| php artisan about | Displays environment, Laravel version, and PHP info |
| Command | Description |
| php artisan list | Lists all available Artisan commands |
| php artisan help migrate | Shows help for a specific command |

**Laravel Blade Conditional & Include Directives**

| **Directive** | **Description / Usage** | **Example** |
| --- | --- | --- |
| @if | Standard conditional check (like if in PHP). | @if($user->isAdmin()) Welcome, Admin! @endif |
| @elseif | Used after an @if condition. | @elseif($user->isEditor()) |
| @else | Executes if previous conditions are false. | @else Welcome, Guest! @endif |
| @endif | Closes an @if block. | — |
| @unless | The inverse of @if — runs if condition is **false**. | @unless($user->isAdmin()) You’re not admin @endunless |
| @isset | Checks if a variable is set (not null). | @isset($name) Hello, {{ $name }} @endisset |
| @empty | Checks if a variable is empty. | @empty($records) No records found! @endempty |
| @switch | Starts a switch-case block. | @switch($role) |
| @case | Defines a case inside a switch. | @case('admin') ... @break |
| @break | Breaks from a @case. | @break |
| @default | Default case inside a switch. | @default No role found @break |
| @endswitch | Ends a @switch block. | — |
| @auth | Checks if user is authenticated. | @auth Welcome, {{ auth()->user()->name }} @endauth |
| @guest | Checks if no user is authenticated. | @guest Please login @endguest |
| @hasrole *(Spatie roles package)* | Checks if user has a specific role. | @hasrole('admin') Admin Panel @endhasrole |
| @can | Checks if user has a specific permission or ability (Gate). | @can('delete-post') <button>Delete</button> @endcan |
| @cannot | Opposite of @can. | @cannot('edit-post') You can’t edit this @endcannot |
| @elsecan | Alternative ability check. | @elsecan('update-post') |
| @env | Runs content if in specific environment (local, production, etc.). | @env('local') Debug mode active @endenv |
| @production | Runs only in production environment. | @production Live site @endproduction |
| @include | Includes a Blade view unconditionally. | @include('partials.header') |
| @includeIf | Includes a view **only if it exists**. | @includeIf('partials.missing') |
| @includeWhen | Includes a view **when a condition is true**. | @includeWhen($user->isAdmin(), 'partials.admin') |
| @includeUnless | Includes a view **when a condition is false**. | @includeUnless($user->isGuest(), 'partials.user') |
| @includeFirst | Includes the **first existing view** from a list. | @includeFirst(['custom.header', 'default.header']) |
| @once | Ensures the code inside runs only once, even in loops. | @once <script>init();</script> @endonce |
| @push / @stack | Push and render sections dynamically (often for scripts). | @push('scripts') <script></script> @endpush and @stack('scripts') |
| @hasSection | Checks if a section exists in the layout. | @hasSection('title') <h1>@yield('title')</h1> @endif |
| @sectionMissing | Checks if a section is missing. | @sectionMissing('content') Default content @endif |
| @forelse | Like @foreach, but includes an “empty” case. | @forelse($users as $user) {{ $user->name }} @empty No users @endforelse |
| @continue | Skips current loop iteration. | @continue($user->banned) |
| @break *(in loops)* | Exits loop early. | @break($loop->iteration == 3) |