

# Laravel Quick Reference

Artisan, routing, Eloquent, Blade, middleware, auth

## Artisan

### Common Commands

<code>php artisan serve</code>	Start development server
<code>php artisan make:model Name -m</code>	Create model with migration
<code>php artisan make:controller NameController</code>	Create controller class
<code>php artisan make:middleware Name</code>	Create middleware class
<code>php artisan migrate</code>	Run pending migrations
<code>php artisan migrate:rollback</code>	Rollback last migration batch
<code>php artisan db:seed</code>	Run database seeders
<code>php artisan tinker</code>	Interactive REPL for your app
<code>php artisan route:list</code>	List all registered routes
<code>php artisan cache:clear</code>	Clear the application cache
<code>php artisan config:clear</code>	Clear cached config
<code>php artisan queue:work</code>	Start processing queued jobs

## Routing

### Basic Routes

```
Route::get('/users', [UserController::class, 'index']);
Route::post('/users', [UserController::class, 'store']);
Route::put('/users/{id}', [UserController::class, 'update']);
Route::delete('/users/{id}', [UserController::class, 'destroy']);
```

### Route Parameters & Groups

```
Route::get('/user/{id}', function (int $id) {
    return User::findOrFail($id);
});

Route::prefix('api')->middleware('auth')->group(function () {
    Route::get('/profile', [ProfileController::class, 'show']);
});
```

### Route Features

<code>-&gt;name('route.name')</code>	Named route for URL generation
<code>-&gt;where('id', '[0-9]+')</code>	Regex constraint on parameter
<code>Route::resource()</code>	RESTful resource routes (7 routes)
<code>Route::apiResource()</code>	API resource (no create/edit views)
<code>Route::fallback()</code>	Catch-all for unmatched routes

## Controllers

### Resource Controller

```
class PostController extends Controller {
    public function index() {
        return view('posts.index', ['posts' => Post::all()]);
    }

    public function store(Request $request) {
        $validated = $request->validate(['title' => 'required|
max:255']);
        Post::create($validated);
        return redirect()->route('posts.index');
    }
}
```

### Resource Methods

<code>index()</code>	GET /resource -- list all
<code>create()</code>	GET /resource/create -- show form
<code>store()</code>	POST /resource -- save new
<code>show(\$id)</code>	GET /resource/{id} -- display one
<code>edit(\$id)</code>	GET /resource/{id}/edit -- edit form
<code>update(\$id)</code>	PUT /resource/{id} -- update
<code>destroy(\$id)</code>	DELETE /resource/{id} -- remove

## Blade Templates

### Layout & Sections

```
{{!-- layouts/app.blade.php --}}
<html><body>
    @yield('content')
</body></html>

{{!-- pages/home.blade.php --}}
@extends('layouts.app')
@section('content')
    <h1>Home</h1>
@endsection
```

### Directives

<code>{{ \$var }}</code>	Echo with HTML escaping
<code>{!! \$html !!}</code>	Echo raw (unescaped)
<code>@if / @elseif / @else</code>	Conditional blocks
<code>@foreach (\$items as \$item)</code>	Loop through collection
<code>@forelse / @empty</code>	Loop with empty fallback
<code>@include('partial')</code>	Include another Blade view
<code>@component / @slot</code>	Reusable Blade components
<code>@csrf</code>	CSRF token hidden field
<code>@auth / @guest</code>	Check authentication status
<code>@error('field')</code>	Display validation error

## Eloquent ORM

### Model Basics

```
class Post extends Model {
    protected $fillable = ['title', 'body', 'user_id'];

    public function user() {
        return $this->belongsTo(User::class);
    }
}
```

### Querying

```
Post::all(); // all records
Post::find(1); // by primary key
Post::where('status', 'published')->get();
Post::where('views', '>', 100)->orderBy('created_at', 'desc')->first();
```

## CRUD Operations

```
$post = Post::create(['title' => 'New', 'body' => '...']);
$post->update(['title' => 'Updated']);
$post->delete();
Post::destroy([1, 2, 3]); // delete by IDs
```

## Relationships

<code>hasOne</code>	One-to-one (User -> Phone)
<code>hasMany</code>	One-to-many (Post -> Comments)
<code>belongsTo</code>	Inverse of hasOne/hasMany
<code>belongsToMany</code>	Many-to-many with pivot table
<code>hasManyThrough</code>	Has-many via intermediate model

## Migrations

### Creating Tables

```
Schema::create('posts', function (Blueprint $table) {
    $table->id();
    $table->foreignId('user_id')->constrained()->cascadeOnDelete();
    $table->string('title');
    $table->text('body')->nullable();
    $table->timestamps();
});
```

### Column Types

<code>\$table-&gt;id()</code>	Auto-increment BIGINT primary key
<code>\$table-&gt;string('col', 100)</code>	VARCHAR with optional length
<code>\$table-&gt;text('col')</code>	TEXT column
<code>\$table-&gt;integer('col')</code>	INTEGER column
<code>\$table-&gt;boolean('col')</code>	BOOLEAN column
<code>\$table-&gt;json('col')</code>	JSON column
<code>\$table-&gt;timestamp('col')</code>	TIMESTAMP column
<code>\$table-&gt;timestamps()</code>	created_at and updated_at
<code>\$table-&gt;softDeletes()</code>	deleted_at for soft deletes

## Middleware

### Custom Middleware

```
class EnsureAdmin {
    public function handle(Request $request, Closure $next) {
        if (!$request->user()->is_admin) {
            abort(403);
        }
        return $next($request);
    }
}
```

### Registering & Using

```
// bootstrap/app.php
->withMiddleware(function (Middleware $middleware) {
    $middleware->alias(['admin' => EnsureAdmin::class]);
});
```

```
// In routes
Route::get('/admin', fn() => '...')->middleware('admin');
```

### Built-in Middleware

<code>auth</code>	Require authentication
<code>guest</code>	Redirect if authenticated
<code>throttle:60,1</code>	Rate limit (60 req/min)
<code>verified</code>	Require email verification
<code>signed</code>	Validate signed URL

# Laravel Quick Reference

## Authentication

### Auth Helpers

```
Auth::check(); // is user logged in?
Auth::user(); // current User model
Auth::id(); // current user ID
Auth::attempt(['email' => $e, 'password' => $p]);
Auth::logout();
```

### Starter Kits

<b>Laravel Breeze</b>	Minimal auth scaffolding (Blade or Inertia)
<b>Laravel Jetstream</b>	Full-featured (teams, 2FA, API tokens)
<b>Sanctum</b>	SPA / mobile API token authentication
<b>Passport</b>	Full OAuth2 server implementation

### Protecting Routes

```
Route::middleware('auth')->group(function () {
    Route::get('/dashboard', [DashController::class, 'index']);
});
```

## Validation

### Controller Validation

```
$validated = $request->validate([
    'title' => 'required|string|max:255',
    'email' => 'required|email|unique:users',
    'age' => 'nullable|integer|min:0',
]);
```

### Form Request

```
class StorePostRequest extends FormRequest {
    public function rules(): array {
        return [
            'title' => 'required|max:255',
            'body' => 'required|min:10',
        ];
    }
}
```

### Common Rules

<b>required</b>	Field must be present and not empty
<b>string   integer   boolean</b>	Type validation
<b>min:N   max:N</b>	Min/max length or value
<b>email</b>	Valid email format
<b>unique:table,column</b>	Must be unique in DB table
<b>exists:table,column</b>	Must exist in DB table
<b>in:a,b,c</b>	Must be one of listed values
<b>confirmed</b>	Requires matching <code>_confirmation</code> field
<b>date   after:date</b>	Date validation

## Common Patterns

### API Response

```
return response()->json(['data' => $users], 200);
return response()->json(['error' => 'Not found'], 404);
```

### Environment & Config

```
env('APP_KEY'); // read .env value
config('app.name'); // read config value
config(['app.debug' => true]); // set at runtime
```

## Useful Helpers

<b>route('name', \$params)</b>	Generate URL for named route
<b>redirect()-&gt;route('name')</b>	Redirect to named route
<b>back()-&gt;withErrors()</b>	Redirect back with validation errors
<b>abort(404)</b>	Throw HTTP exception
<b>collect(\$array)</b>	Create a Collection from array
<b>now()</b>	Current Carbon datetime
<b>cache()-&gt;remember()</b>	Cache a value with TTL