

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	
<code>Route::get('/users', [UserController::class, 'index']);</code>	
<code>Route::post('/users', [UserController::class, 'store']);</code>	
<code>Route::put('/users/{id}', [UserController::class, 'update']);</code>	
<code>Route::delete('/users/{id}', [UserController::class, 'destroy']);</code>	
Route Parameters & Groups	
<code>Route::get('/user/{id}', function (int \$id) { return User::findOrFail(\$id); });</code>	
<code>Route::prefix('api')->middleware('auth')->group(function () { Route::get('/profile', [ProfileController::class, 'show']); });</code>	
Route Features	
<code>->name('route.name')</code>	Named route for URL generation
<code>->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	
<pre>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'); } }</pre>	
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	
<pre>{{-- layouts/app.blade.php --}} <html><body> @yield('content') </body></html> {{-- pages/home.blade.php --}} @extends('layouts.app') @section('content') <h1>Home</h1> @endsection</pre>	
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	
<pre>class Post extends Model { protected \$fillable = ['title', 'body', 'user_id']; public function user() { return \$this->belongsTo(User::class); } }</pre>	
Querying	
<code>Post::all();</code>	// all records
<code>Post::find();</code>	// by primary key
<code>Post::where('status', 'published')->get();</code>	
<code>Post::where('views', '>', 100)->orderBy('created_at', 'desc')->first();</code>	

CRUD Operations	
<code>\$post = Post::create(['title' => 'New', 'body' => '...']);</code>	
<code>\$post->update(['title' => 'Updated']);</code>	
<code>\$post->delete();</code>	
<code>Post::destroy(1, 2, 3);</code>	// 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	
<pre>Schema::create('posts', function (Blueprint \$table) { \$table->id(); \$table->foreignId('user_id')->constrained()->cascadeOnDelete(); \$table->string('title'); \$table->text('body')->nullable(); \$table->timestamps(); });</pre>	
Column Types	
<code>\$table->id()</code>	Auto-increment BIGINT primary key
<code>\$table->string('col', 100)</code>	VARCHAR with optional length
<code>\$table->text('col')</code>	TEXT column
<code>\$table->integer('col')</code>	INTEGER column
<code>\$table->boolean('col')</code>	BOOLEAN column
<code>\$table->json('col')</code>	JSON column
<code>\$table->timestamp('col')</code>	TIMESTAMP column
<code>\$table->timestamps()</code>	created_at and updated_at
<code>\$table->softDeletes()</code>	deleted_at for soft deletes

Middleware	
Custom Middleware	
<pre>class EnsureAdmin { public function handle(Request \$request, Closure \$next) { if (!\$request->user()?->is_admin) { abort(403); } return \$next(\$request); } }</pre>	
Registering & Using	
<code>// bootstrap/app.php</code>	
<code>->withMiddleware(function (Middleware \$middleware) { \$middleware->alias(['admin' => EnsureAdmin::class]); })</code>	
<code>// In routes</code>	
<code>Route::get('/admin', fn() => '...')->middleware('admin');</code>	

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

Starter Kits	
<code>Laravel Breeze</code>	Minimal auth scaffolding (Blade or Inertia)
<code>Laravel Jetstream</code>	Full-featured (teams, 2FA, API tokens)
<code>Sanctum</code>	SPA / mobile API token authentication
<code>Passport</code>	Full OAuth2 server implementation
Protecting Routes	
<code>Route::middleware('auth')->group(function () { Route::get('/dashboard', [DashboardController::class, 'index']); });</code>	
Validation	
Controller Validation	
<pre>\$validated = \$request->validate(['title' => 'required string max:255', 'email' => 'required_email unique:users', 'age' => 'nullable integer min:0',]);</pre>	

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

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

Common Patterns	
API Response	
<code>return response()->json(['data' => \$users], 200);</code>	
<code>return response()->json(['error' => 'Not found'], 404);</code>	
Environment & Config	
<code>env('APP_KEY');</code>	// read .env value
<code>config('app.name');</code>	// read config value
<code>config(['app.debug' => true]);</code>	// set at runtime
Useful Helpers	
<code>route('name', \$params)</code>	Generate URL for named route
<code>redirect()->route('name')</code>	Redirect to named route
<code>back()->withErrors()</code>	Redirect back with validation errors
<code>abort(404)</code>	Throw HTTP exception
<code>collect(\$array)</code>	Create a Collection from array
<code>now()</code>	Current Carbon datetime
<code>cache()->remember()</code>	Cache a value with TTL