

Q1. What is Laravel's request lifecycle from incoming HTTP to response?

A: HTTP request enters public/index.php, bootstraps the service container, passes through HTTP Kernel middleware (CORS, session, auth), dispatches to the Router, which resolves the controller method and returns a Response that Laravel sends back.

Q2. How does Laravel Eloquent ORM define model relationships?

A: Define a public method in the model. hasMany(Order::class) returns a HasMany relation (posts for a user). belongsTo(User::class) defines the inverse. Eager-load with User::with('orders')->find(1) to avoid N+1. ManyToMany uses a pivot table via belongsToMany.

Q3. How does Laravel's service container resolve dependencies?

A: When Laravel constructs a controller, it reads its constructor type hints and resolves each from the container. Bind an interface: \$this->app->bind(PaymentGateway::class, StripeGateway::class). All classes are resolved recursively through the container.

Q4. What is Laravel Artisan and what are five common commands?

A: php artisan make:model Order -m (model + migration), make:controller (controller), migrate (apply migrations), queue:work (start queue worker), route:list (print all routes), tinker (REPL against the app). Custom commands extend Illuminate\Console\Command.

Q5. How does Laravel handle queued jobs?

A: Implement ShouldQueue on a Job class. Dispatch with SendEmail::dispatch(\$user). A queue worker (php artisan queue:work) fetches jobs from the backend (Redis, SQS, DB) and executes them. Failed jobs are stored in failed_jobs and can be retried with queue:retry all.

Q6. What is Laravel Sanctum and how does it authenticate SPAs?

A: Sanctum issues API tokens stored in the personal_access_tokens table. For SPAs it uses cookie-based session auth with CSRF protection via the /sanctum/csrf-cookie endpoint. Token-based auth is for mobile clients. Both use the Auth::user() facade.

Q7. How does Laravel Blade template inheritance work?

A: @extends('layouts.app') references the parent template. @section('content') and @endsection define overrideable regions. @yield('content') in the parent marks where child content renders. @include('partials.nav') embeds a partial. @stack/@push build asset

Q8. How do you write feature tests in Laravel?

A: Subclass Tests\TestCase and use the RefreshDatabase trait to reset DB per test. \$response = \$this->postJson('/api/login', ['email'=>..., 'password'=>...]). Assert \$response->assertStatus(200)->assertJsonStructure(['token']). Use

Q9. What are Laravel Policies and how do they control authorization?

A: A Policy class has methods like view(User \$user, Post \$post). Register in AuthServiceProvider. Call \$this->authorize('view', \$post) in a controller. Laravel resolves the policy and throws 403 on failure. Gate::allows('view', \$post) checks without throwing.

Q10. What is Laravel Livewire and how does it work?

A: Livewire is a full-stack component framework. A PHP class (extends LivewireComponent) holds state and methods. The Blade template uses wire:model for two-way binding and wire:click for actions. AJAX requests sync state between browser and server on each interaction.