



PHP and Laravel Training Course: Eloquent ORM, REST APIs and PHPUnit Testing



PHP and Laravel Training Course: Eloquent ORM, REST APIs and PHPUnit Testing

Introduction:

PHP and Laravel training on Eloquent ORM, REST APIs and PHPUnit testing is a 5-day course for PHP web developers and maintainers of server-rendered sites, ending with a tested Laravel web application that exposes a token-protected REST API. Many organisations still run PHP sites built from hand-written SQL, copied include files and untested forms, so each change risks broken pages, leaked records and slow releases. Nominees already write PHP or another server-side language at work, and every day runs as a hands-on lab with Composer, Artisan and a MySQL database. CoreConcept Training Center delivers this PHP and Laravel course.

Course Objectives:

- Write modern PHP using typed properties, enums, readonly classes, traits, namespaces and PSR-4 autoloading managed through Composer
- Organise a Laravel project with Artisan, environment files, named routes, middleware, resource controllers and Blade layouts and components
- Model MySQL data with migrations, factories, seeders and Eloquent relationships, preventing N+1 queries and mass assignment flaws
- Validate form input with Form Requests and control sign-in and record access with session guards, gates and policies
- Publish JSON endpoints with API Resources and Sanctum token abilities, offload slow work to queued jobs and the task scheduler, and verify behaviour with PHPUnit feature tests
- Deploy a tested Laravel web application with its REST API, PHPUnit suite and production configuration as a reference project

Target Audience:

- PHP developers responsible for building and extending business websites and portals
 - Web programmers responsible for maintaining older procedural PHP code and moving it onto a framework
 - Back-end developers responsible for JSON endpoints consumed by mobile or single-page front ends
 - Web team leads responsible for code review, package choices and release quality in PHP teams
 - Full-time web support staff responsible for MySQL-backed forms, user accounts and scheduled data tasks
-

Course Outline:

Day 1: Modern PHP Language, Object-Oriented Code and Composer Packages

- PHP Typed Properties, Enums, Readonly Classes and Match Expressions
- Classes, Interfaces and Traits for Reusable PHP Domain Logic
- PHP Namespaces, Use Imports and PSR-4 Autoloading Rules
- Composer Require Constraints, Lock File and Package Updates
- PHP Exceptions, Closures and Arrow Functions in Service Classes

Day 2: Laravel Project Structure, Routing, Controllers, Blade Views and Sign-In

- Laravel Directory Layout, Artisan Commands and Environment Configuration Files
- Route Definitions, Named Routes, Route Groups and Middleware
- Resource Controllers, Route Model Binding and Redirect Responses
- Blade Layouts, Components, Slots and Loop Directives
- Starter Kit Login, Password Hashing and Session Guards

Day 3: Eloquent ORM, MySQL Migrations, Seeders and Form Validation

- Migrations, Schema Builder Columns, Indexes and Foreign Keys on MySQL
- Model Factories and Database Seeders for Repeatable Test Data
- Eloquent Models, Mass Assignment Guards and Attribute Casting
- Eloquent Relationships, Eager Loading and N+1 Query Prevention
- Form Request Validation Rules, Error Bags and Old Input

Day 4: Authorisation, REST APIs, Queues, Security Controls and PHPUnit Testing

- Gates and Policies for Record-Level Authorisation Decisions
- API Resources, JSON Responses and Sanctum Token Abilities
- Queued Jobs, Failed Job Retries and Task Scheduler Entries
- CSRF Tokens, Output Escaping, Rate Limiting and Encrypted Secrets
- PHPUnit Unit Tests and HTTP Feature Tests with RefreshDatabase

Day 5: Lab Building, Testing and Deploying a Laravel Application with an API

- Capstone Brief, Database Schema and Route Map Definition
 - Lab Implementation of Eloquent Models, Blade Pages and Policies
 - Lab Build of Sanctum-Protected API Endpoints and Queued Notifications
 - Lab Deployment with Config Caching, Queue Workers and Scheduler Cron
 - Tested Laravel Application Completion with PHPUnit Run and Code Review
-

Skills You Will Gain:

- Modern PHP Syntax
- Composer Dependency Management
- Laravel Routing and Middleware
- Blade Template Development
- Eloquent Data Modelling
- Policy-Based Access Control
- JSON Endpoint Development
- PHPUnit Feature Testing

Why Attend This Course:

- Hand the web team lead a tested Laravel application with its Sanctum-protected API, PHPUnit suite and deployment checklist as the template for new PHP projects
- Choose between Blade pages and JSON endpoints, sessions and API tokens, and synchronous code or queued jobs for each new feature
- Avoid SQL injection, mass assignment leaks, N+1 slowdowns and missed scheduled tasks that cause incidents and rework after release
- Share the migration conventions, policy classes and PHPUnit feature test patterns with other PHP developers on the team

Conclusion:

Back at work, the participant hands the web team lead a tested Laravel application with a Sanctum-protected REST API, migrations, seeders, a PHPUnit feature test suite and a deployment checklist. The lead uses it to decide how legacy PHP sites are moved onto Laravel, how new endpoints are secured and which work runs in queues or on the scheduler. After the first feature built from this reference reaches production, the team should review failing tests, slow query logs, failed jobs and authorisation denials.

Frequently Asked Questions FAQ:

What should participants know before the PHP and Laravel training course?

Participants should already write PHP or another server-side language and understand HTML forms, HTTP requests and basic SQL. A laptop able to run PHP, Composer, a MySQL server and a code editor is used in every daily lab.

How does the PHP and Laravel training course differ from a JavaScript full-stack or API management course?

It builds server-rendered pages and JSON endpoints in PHP with Laravel, Eloquent and PHPUnit. JavaScript full-stack courses use browser frameworks and Node servers, while API management courses cover gateways, developer portals and API governance rather than writing application code.

Why do PHP teams adopt Laravel and Eloquent ORM for web applications?

Laravel supplies routing, Blade templates, migrations, authentication, queues and a scheduler in one framework, and Eloquent maps MySQL tables to PHP classes with relationships, so teams replace hand-written SQL and include files with a consistent, testable structure.

What do participants take back from the PHP and Laravel training course?

Participants take back a working Laravel application with Blade pages, a Sanctum-protected REST API, migrations and seeders, a PHPUnit feature test suite and a deployment checklist covering config caching, queue workers and scheduler entries.