

Updated on 09/24/2026

Sign Up

Inertia.js Training: Performance & Modern Monolith

2 days (14 hours)

Overview

Inertia.js allows you to build applications using a Modern Monolith architecture: the backend handles routing, controllers, authentication, and data access, while the frontend framework provides a user experience similar to a SPA, without requiring a separate API.

Our Inertia.js: Performance & Modern Monolith training course will teach you how to design modern applications using Laravel, Inertia.js, and Vue.js while maintaining a unified architecture between the backend and frontend.

You'll learn how to build reactive pages, pass data using props, manage navigation, forms, and Laravel validation, all while delivering a user experience similar to that of a Single Page Application.

The course will also teach you how to optimize the performance of the Modern Monolith using partial reloads, deferred props, prefetching, code splitting, and optimization of data exchanged between Laravel and Vue.js. You'll also explore server-side rendering to optimize the application's initial load and deployment.

Our hands-on workshop will guide you through the step-by-step process of building a Modern Monolith application, from its Laravel and Vue.js architecture to performance optimization and production readiness.

By the end of this training, participants will be able to design, optimize, and deploy a high-performance Modern Monolith application using Laravel, Inertia.js, and Vue.js.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Understand the Modern Monolith architecture and the role of Inertia.js
- Build an application with Laravel, Inertia.js, and Vue.js without a separate API
- Master Inertia.js pages, props, forms, and navigation mechanisms
- Optimize performance with partial reloads, deferred props, and prefetching
- Implement advanced features and server-side rendering

Target Audience

- Laravel developers
- Vue.js developers
- Full-Stack Developers
- Software Architect

Prerequisites

- Proficiency in the fundamentals of PHP
- Have some experience with the Laravel framework
- Familiarity with the principles of modern JavaScript
- Knowledge of the fundamentals of components and reactivity with Vue.js

Technical Prerequisites

- Development environment with PHP, Composer, and Laravel up and running
- Node.js and npm installed to manage frontend dependencies, Vite, and server-side rendering (SSR)
- Code editor suitable for PHP, JavaScript, and Vue.js development
- A recent web browser with developer tools enabled

Curriculum for our Inertia.js: Performance & Modern Monolith training course

[Day 1 - Morning]

Inertia.js and Modern Monolith architecture

- Understanding how Inertia.js works and its role
- Understanding the concept of the Modern Monolith and its architectural principles
- Compare the Modern Monolith, traditional applications, and SPAs with separate APIs
- Understand Laravel's role in routing, controllers, authentication, and data access
- Use Vue.js to build a reactive user interface without maintaining a dedicated REST API

- Install and configure Inertia.js with Laravel and Vue.js

[Day 1 - Afternoon]

Pages, props, and navigation with Inertia.js

- Create and return pages with `Inertia::render`
- Passing data from the backend to Vue components using props
- Sharing data across multiple pages with Shared Data
- Set up persistent layouts and organize application components
- Manage links and client-side navigation without a full page reload
- Control the size and lifecycle of props to limit unnecessary data transfers

Forms and business logic

- Create and manage Inertia forms
- Leverage Laravel validation in the Vue.js interface
- Display and manage validation errors
- Preserve component state during navigation
- Handling redirects, flash messages, and user interactions
- Minimize unnecessary page reloads to maintain a smooth user experience

[Day 2 - Morning]

Optimize the performance of the Modern Monolith

- Identify data and requests that slow down navigation Inertia.js
- Use Partial Reloads to refresh only the necessary data
- Leverage Deferred Props to defer the loading of non-priority data
- Implement prefetching to anticipate navigation
- Optimize lists, data volumes, and props transmitted by the backend
- Measure the impact of optimizations on load times and navigation

[Day 2 - Afternoon]

Advanced Features and the SPA Experience

- Implement lazy loading and asynchronous interactions
- Manage scrolling, history, and state preservation
- Building interfaces with pagination and infinite scrolling
- Use polling when data requires regular updates
- Implement code splitting to reduce the amount of JavaScript loaded initially
- Maintain an experience similar to that of a SPA while preserving the Modern Monolith architecture

SSR, optimization, and deployment

- Understand how Server-Side Rendering (SSR) works with Inertia.js
- Identify scenarios where SSR improves initial load times and SEO
- Configure server-side rendering with Laravel and Vue.js
- Optimize assets and prepare the application for production
- Monitor frontend and backend performance of the Modern Monolith
- Apply best practices for deploying and maintaining an Inertia.js application

Target Audience

This training is intended for both individuals and companies—large or small—that wish to train their teams in a new, advanced IT technology or to acquire specific domain knowledge or modern methodologies.

Assessment upon enrollment

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency in various types of technologies, as well as their expectations and personal goals for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Hands-On Training: 60% Practical, 40% Theoretical. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the instructor—supported by examples and reflection sessions—and group work.

Assessment

At the end of the session, a multiple-choice quiz is administered to verify that participants have successfully acquired the skills.

Certification

A certificate will be issued to each trainee who has completed the entire training program.