

Nuxtjs Advanced Training

3 days (21 hours)

Presentation

Our advanced Nuxt.js training course will enable you to develop high-performance, modern and universal web applications, by mastering the key concepts of the Nuxt framework and the Vue ecosystem.

Our training program will teach you all the skills you need to operate Nuxt at an advanced level: modular architecture, optimized data fetching, dynamic routing, customized middleware, state with Vuex or Pinia, deployment and securing your applications.

At the end of this course, you'll be able to design a complete Nuxt application, optimized for performance and SEO, and adapted to large-scale projects. You'll also learn how to migrate efficiently from Nuxt 2 to Nuxt 3 and take advantage of the new features in Vue 3 and the Vite/Nitro environment.

As with all our training courses, this one uses the latest version of [Nuxt 3.1](#).

Objectives

- Understanding the universal application concept and its benefits
- Master Nuxt.js and its rendering modes SSR, SSG, SPA, ISR
- Create a complete web application with Nuxt 3
- Know how to manage client- and server-side data
- Implement customized middleware for security and authentication

Target audience

- Web Developer
- Backend developer

- Frontend developer

Prerequisites

- Web development experience (Javascript, CSS, HTML)
- Knowledge of backend development
- Experience with the Vuejs framework
- knowledge of Nuxt JS

Nuxt training program

Introduction to NuxtJs Advanced

- Reminder of concepts :
 - architecture Nuxt
 - SSR vs SSG operation
- NuxtJS news and developments
- Server-side rendering optimization strategies
- Choose between SSR, SSG, ISR, or SPA depending on your needs

Advanced architecture and configuration

- Fine-tuning nuxt.config
- Advanced project modularization
- Structuring a large-scale project
- Organizing complex layouts
- Global vs. specific middleware
- RuntimeConfig optimization

Advanced routing

- Complex routes with multiple parameters
- Conditional, protected or customized routes
- Strategic use of middleware
- Internationalization with Nuxt
- Lazy-loading roads

In-depth view and components

- Composition API vs Options API in a NuxtJS project
- Logic reuse with composables
- Dependency injection with provide/inject
- Advanced slots, v-slots and dynamic components
- Performance management for heavy components

Advanced data fetching

- Managing loading status, errors and fallback
- useFetch, asyncData, useAsyncData, useLazyFetch
- Client-side hydration vs. server-side pre-rendering
- Strategic caching
 - HTTP
 - memory
 - payload
- Integration with RESTful or GraphQL APIs
- Use Nitro to create custom backend endpoints

Authentication & security

- Setting up an authentication system with JWT/OAuth
- Advanced integration of nuxt-auth and auth-module
- SSR/SPA strategies for authentication
- Protection of routes, pages and API endpoints
- Secure token storage

State management Advanced

- Vuex advanced :
 - dynamic modules
 - plugins
 - middleware
- Migrating to pinia for Nuxt 3
- Complex forms management and client/server validation
- Persistence of condition
- Synchronize frontend/backend status

Testing , code quality and CI/CD

- Unit testing with Jest or Vitest
- E2E tests with Cypress or Playwright
- Mocking backend requests with MSW
- Set up ESLint, Prettier, Stylelint automated workflows
- CI/CD with GitHub Actions or GitLab CI
- Performance and error monitoring

Deployment & performance

- Deployment on Vercel, Netlify, Render, or a custom server
- Performance analysis with Lighthouse and Web Vitals
- Optimizing bundle size
- Use serverMiddleware to create a custom API

- Server/API/CDN cache
- Monitoring and observability

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.