

Updated on 02/24/2026

Register

NestJS Training

3 days (21 hours)

Overview

NestJS is a Node.JS framework that is mainly used to create efficient and scalable APIs. Nest.JS has grown enormously in popularity thanks to its excellent features. In 2021, NestJS exploded in popularity with over 40,000 stars on Github. It is easy to use and quick to learn.

This framework uses JavaScript and can also use TypeScript. It combines elements of OOP (object-oriented programming), FP (functional programming), and RFP (reactive functional programming).

The latter also helps you progress by structuring your application correctly. NestJS brings together a set of technologies and features needed to build reliable and durable HTTP servers using Nodejs.

Nest.JS implements the Express framework by default but also allows the use of Fastify. Currently, this framework is growing rapidly on TypeScript in the Node.JS universe. With NestJS, you can write scalable, testable, and loosely coupled applications.

To start a new project on [Node.JS](#), NestJS is an excellent choice because it is based on a modular architecture. This allows you to define controllers, services, middleware, pipes, and guards within them, similar to Angular, from which it draws inspiration.

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

Objectives

- Learn how to master the NestJS platform
- Learn how to create applications on NestJS
- Create a progressive architecture for large applications

Target audience

Web developers

Prerequisites

- Proficiency in JavaScript
- Knowledge of Node.JS
- Knowledge of TypeScript basics is a plus

Technical requirements

- NodeJS version 16+ installed
- Docker and Git installed
- **Visual Studio Code** or another text editor

Our NestJS training program

[Day 1 - Morning]

Fundamentals and Revolution NestJS 12

- Understanding the NestJS philosophy: modularity and dependency injection (DI)
- Discover what's new in v12: SWC build engine and native ESM support
- Installing and getting started with the NestJS CLI
- Anatomy of a project: Modules, Controllers, and Providers
- Lifecycle of a NestJS application (Request Lifecycle)
- Hands-on workshop: Initializing a project and creating a modular API.

[Day 1 - Afternoon]

Mastering business logic and validation

- Using Middleware and Interceptors
- Data validation with Pipes and class-validator
- Global error management via Exception Filters
- Dynamic configuration with `@nestjs/config` and schema validation (Zod/Joi)
- Performance optimization with Fastify vs Express

- Hands-on workshop: Setting up a robust logging and validation system.

[Day 2 - Morning]

Data persistence and advanced typing

- Integration of modern ORMs: Prisma or TypeORM
- Architecture patterns: Repository and Data Mapper
- Database migration and seeding management
- Handling complex relationships and transaction management
- Optimizing data access (Lazy loading vs. Eager loading)
- Hands-on workshop: Connecting to PostgreSQL and modeling a relational database.

[Day 2 - Afternoon]

Security and Authentication (Modern Stack)

- Implementing Guards and Passport.js strategies
- JWT (JSON Web Tokens) authentication and refresh tokens
- Authorization management: RBAC (Roles) and ABAC (Attributes)
- Protection against vulnerabilities (CORS, Helmet, Rate-limiting)
- Securing secrets with environment variables
- Hands-on workshop: Implementation of a complete and secure authentication system.

[Day 3 - Morning]

Distributed Architectures & Microservices

- Introduction to Microservices with NestJS
- Transport Protocols: Redis, RabbitMQ, or gRPC
- Asynchronous communication and event patterns
- Queue Management with BullMQ
- Hybrid architecture: API Gateway and microservices
- Hands-on workshop: Creating a decoupled asynchronous processing service.

[Day 3 - Afternoon]

Industrialization, testing, and deployment

- Unit and integration testing with Jest
- End-to-end (e2e) testing with Supertest
- Observability: Monitoring with Terminus and OpenTelemetry
- Containerization with Docker (multi-stage optimization)
- CI/CD deployment strategies and production checklist

- Hands-on workshop: Final project integrating testing, Docker, and monitoring.

Target companies

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

Positioning at the start of training

The positioning at the start of the training complies with Qualiopi quality criteria. Upon final registration, the learner receives a self-assessment questionnaire that allows us to assess their estimated level of proficiency in different types of technologies, as well as their expectations and personal objectives 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 be problematic for the monitoring and smooth running of the training session.

Teaching methods

Practical training: 60% practical, 40% theory. Training materials distributed in digital format to all participants.

Organization

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

Assessment

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been correctly acquired.

Certification

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