



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Node.js Backends with TypeScript

Class Duration

21 hours of live training delivered over 3 days.

Student Prerequisites

- Working TypeScript knowledge (TypeScript Essentials or equivalent)
- Professional JavaScript or TypeScript development experience
- Basic familiarity with HTTP and REST concepts
- Basic SQL knowledge helpful but not required

Target Audience

This course is designed for developers building server-side applications with Node.js and TypeScript. It suits front-end developers moving into full-stack work, backend developers arriving from other platforms, and teams standardizing on a modern TypeScript backend stack who want current, production-oriented guidance on frameworks, validation, data access, testing, and deployment.

Description

Node.js Backends with TypeScript teaches participants to design, build, test, and ship production-grade backend services using the current TypeScript-first Node.js stack. The course starts with modern project setup — ES modules, fast iteration with `tsx`, Node's now-stable native type stripping, and production builds with `tsup/esbuild` — then builds real HTTP services with Fastify 5 as the primary framework, including a practical comparison with Express 5 so teams can make an informed choice. Runtime validation with Zod 4 anchors a schema-first approach to DTOs, ensuring the types the compiler sees match the data the server actually receives.

From there the course assembles the rest of a production service: layered architecture with dependency injection, database access with Drizzle (with a comparison to Prisma 7), authentication with JWT and OIDC, and end-to-end type safety using tRPC 11 and OpenAPI type generation. Participants implement structured error handling, logging with `pino`, and a test suite combining Vitest 4 unit tests with supertest-style integration tests against real route handlers. The course closes with Docker deployment and



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

observability hooks using OpenTelemetry — covered here as integration points, with full instrumentation depth available in our dedicated Observability with OpenTelemetry course.

Learning Outcomes

- Set up a modern Node.js + TypeScript project with ESM, tsx for development, and tsup/esbuild for production builds
- Explain Node's native type stripping and decide when to run TypeScript directly versus building
- Build HTTP APIs with Fastify 5 using routes, plugins, hooks, and schema-based serialization
- Compare Fastify and Express 5 and choose appropriately for a given project
- Validate all inbound data with Zod 4 schemas and derive static DTO types from them
- Structure services with layered architecture and lightweight dependency injection
- Implement type-safe database access with Drizzle, including migrations and transactions
- Secure APIs with JWT-based authentication and OIDC integration
- Deliver end-to-end type safety with tRPC 11 and generate types from OpenAPI specifications
- Implement consistent error handling and structured logging with pino
- Test services with Vitest 4 unit tests and supertest-style integration tests
- Containerize and deploy a Node.js service with Docker
- Add OpenTelemetry hooks for traces, metrics, and logs

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

Software Requirements

- Node.js 24 LTS
- Visual Studio Code or another TypeScript-aware editor
- Docker Desktop or Podman for container labs
- Permission to install npm packages and editor extensions
- Git and a free GitHub account for lab repositories



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Training Topics

Modern Node.js + TypeScript Setup

- ES modules in Node.js: `package.json`, exports, and import rules
- Fast development iteration with `tsx`
- Node's native type stripping: running `.ts` files directly and its limits
- Production builds with `tsup` and `esbuild`
- `tsconfig` for backend services
- Environment configuration and secrets handling

Fastify Fundamentals

- Fastify 5 architecture: routes, plugins, encapsulation
- Request lifecycle and hooks
- Schema-based validation and fast JSON serialization
- Decorators and typed request/reply
- The plugin ecosystem

Express 5 Comparison

- What changed in Express 5: promises, error handling, router updates
- Middleware model versus Fastify's plugin model
- Performance and ecosystem trade-offs
- Choosing a framework for your team

Runtime Validation with Zod

- Why compile-time types are not enough at the network boundary
- Zod 4 schemas, parsing, and error reporting
- Schema-first DTOs: deriving types with `z.infer`
- Composing and refining schemas
- Integrating Zod with Fastify route validation

Layered Architecture and Dependency Injection

- Routes, services, and repositories
- Constructor injection and composition roots
- Interfaces for testability
- Keeping framework code at the edges

Database Access with Drizzle

- Schema definition in TypeScript
- Type-safe queries, joins, and transactions



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Migrations with drizzle-kit
- Connection pooling and lifecycle
- Comparison: Prisma 7 and its TypeScript-native client

Authentication and Authorization

- JWT structure, signing, and verification
- Access and refresh token patterns
- OIDC integration with an identity provider
- Route-level authorization with Fastify hooks
- Common token-handling pitfalls

End-to-End Type Safety

- tRPC 11: routers, procedures, and client inference
- When tRPC fits and when REST is the right call
- OpenAPI: generating specs from schemas
- Generating client types from OpenAPI documents
- Sharing types across a monorepo

Error Handling

- Error taxonomies: expected versus unexpected failures
- Centralized error mapping to HTTP responses
- Problem details and consistent error payloads
- Avoiding leaky internal errors

Logging with pino

- Structured JSON logging
- Request-scoped child loggers
- Redaction of sensitive fields
- Log levels and transport configuration

Testing

- Vitest 4 setup for backend projects
- Unit testing services with mocked repositories
- Integration tests with supertest-style requests against Fastify
- Test databases and data management
- Coverage and CI integration

Docker Deployment

- Multi-stage Dockerfiles for Node.js + TypeScript



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Small, secure production images
- Health checks and graceful shutdown
- Container configuration via environment

Observability Hooks

- OpenTelemetry concepts: traces, metrics, logs
- Auto-instrumentation for Node.js, Fastify, and database clients
- Exporting to a collector
- Going deeper: our dedicated Observability with OpenTelemetry course