



NestJS Intensiv-Schulung

Link zu dieser Schulung: workshops.de/seminare-schulungen-kurse/nestjs

Über die NestJS Intensiv-Schulung

NestJS ist ein modernes Node.js-Framework für den Aufbau effizienter und skalierbarer serverseitiger Anwendungen. Das Framework kombiniert objektorientierte, funktionale und reaktive Programmierparadigmen. NestJS verwendet TypeScript als Standard, ermöglicht aber auch reines JavaScript. Mit seinem modularen System bietet es eine flexible Architektur, die das Einbinden verschiedener Bibliotheken oder Tools erleichtert. Es ist ideal für die Entwicklung von modernen Microservice-Strukturen und Webanwendungen.

Schulungsformen

Inhouse Schulung

- Termin nach Wunsch
- 3 ganze Tage oder 6 halbe Tage
- Vor Ort, als auch Remote
- Deutsch/Englisch möglich

Öffentliche Schulung

- 5 Termine verfügbar
- 6 halbe Tage
- Remote
- Kurssprache Deutsch

Das bringt dir die NestJS Intensiv-Schulung

Als Entwickler:in

- **Zeit sparen:** Vermeide langes Debugging, indem du die Kernelemente und Prinzipien von NestJS von Grund auf verstehst.
- **Effizientes Lernen:** Verbringe nicht Wochen damit, dir Informationen über NestJS zusammenzusuchen, wenn du das Wichtigste in nur wenigen Tagen lernen kannst.
- **Interaktives Lernen:** Nicht jeder zieht Bücher dem lebendigen Austausch vor. Bei uns ist das Lernen hands-on, interaktiv und macht Spaß.
- **Netzwerken:** Tausche dich mit anderen Teilnehmern aus und erweitere so deinen beruflichen Horizont.
- **Steigere deinen Marktwert:** Mit dem Trend zu modernen Back-End-Frameworks ist das Wissen um NestJS gefragter denn je. Bleibe up-to-date und mache dich unverzichtbar.
- **Persönliche Fragen:** Hast du spezifische Fragen, die du nicht einfach googlen kannst? Bei uns bekommst du die Antworten von Experten.
- **Best Practices:** NestJS hat seine eigenen Konventionen und Best Practices. Wir stellen sicher, dass du diese von Anfang an richtig anwendest.

Das solltest du mitbringen

- **Grundlagenkenntnisse** in [JavaScript](#) und [TypeScript](#)
- **Technische Voraussetzung:** Notebook mit Windows, Linux oder macOS
- Ein grundlegendes Verständnis von Back-End-Entwicklung ist von Vorteil, aber nicht zwingend erforderlich.

Zielgruppe

Dieser Kurs richtet sich an **Einsteiger**.

Als Entscheider:in

- **Richtige Technologieauswahl:** Du planst ein neues Projekt? Mit unserer Schulung kannst du sicherstellen, dass NestJS das passende Framework für eure Anforderungen ist.
- **Kosteneffizienz:** Fehler zu Projektbeginn können teuer sein. Unsere Schulung hilft, typische Anfängerfehler zu vermeiden und somit Kosten zu sparen.

- **Wettbewerbsvorteil:** Mit der rasanten Entwicklung im Technologiebereich ist es wichtig, immer einen Schritt voraus zu sein. Mit NestJS bist du bestens für moderne und skalierbare Anwendungen gerüstet.
- **Schneller Markteinstieg:** Warum Wochen oder Monate mit Selbststudium verbringen? Nach unserer Intensivschulung kann dein Team direkt mit der professionellen Entwicklung in NestJS starten.
- **Erfahrungswissen:** Profitiere von unserer Erfahrung. Wir haben die Stolpersteine von NestJS bereits für dich aus dem Weg geräumt und teilen unsere Erkenntnisse, damit du diese Fehler vermeiden kannst.

Mit der NestJS Intensiv-Schulung bist du bestens gerüstet, um moderne, effiziente und skalierbare Back-End-Lösungen zu entwickeln. Werde zum NestJS-Profi und setze neue Maßstäbe in deinen Projekten!

Themen und Agenda - Beispiel

Dies ist unsere empfohlene Agenda für das Thema **NestJS**. Wir verfügen darüber hinaus über weiteres Kursmaterial, um Themen einfach auszutauschen.

Grundlagen TypeScript

Wir starten mit einer kurzen Einführung in TypeScript (ca. 3 Stunden). Dort lernen wir die Grundlagen für den anschließenden Nest-Teil.

- Unterschiede zu ES6
- Funktionen und Scopes
- Fat Arrow
- Typen
- Klassen
- Decorators / Annotationen
- Modul-System
- Interfaces
- Generics
- Destructuring

Grundlagen von NestJS

Was ist NestJS und aus welche Bestandteilen besteht eine Anwendung die hiermit geschrieben ist?

- Die Nest CLI

- Controller
- Dependency Injection
- Provider, Services
- Module
- NestFactory
- Lifecycle

Architektur großer Anwendungen

Wie teilt man eine Applikation am Besten auf und warum? Wie programmiere ich zukunftssicher? In diesem Teil liefern wir die Antworten.

- Schneiden von Modulen
- Wiederverwendbare Pakete
- Arbeiten mit Monorepos
- Microservices

Anbindung an eine Datenbank

Wie binde ich eine Datenbank an? Wie nutze ich TypeORM mit dem Repository-Pattern? Wie teste ich Services die eine Datenbank nutzen? Diese und andere Fragen beantworten wir in diesem Modul.

- TypeOrm
- Repository Pattern
- Anbinden mehrerer Datenbanken
- MySQL, Postgres, MongoDB
- Testen und Mocken

Testing & Debugging

Mit dem Wachsen einer Applikation wird das manuelle Testen immer aufwändiger bis fast unmöglich. Wir zeigen - wann man welche Tests einsetzt, wie man testet und Best Practices.

- Unit Tests vs. End-to-End Tests
- Unit Testing
- End-To-End Testing mit Supertest
- Integration in CI Umgebungen

REST & GraphQL

Ihr benutzt Nest um eine API-Endpoint zu implementieren? Wir schauen uns sowohl Rest als auch GraphQL als mögliche Protokolle an und gehen auf Vor- und Nachteile ein.

- Einführung

- REST vs GraphQL
- CRUD Endpoints
- GraphQL Resolver, Mutations

Deployment & Performance

Der Titel spricht für sich. Wie deploy ich meine Anwendung in Production ... mit Docker? Wie geh ich mit verschiedenen Enviornments um? Wie optimiere ich die Ladezeit meine Anwendung?

- Build with Docker
- Terminus (Kubernetes readiness / liveness checks)
- Umgang mit Environment Variablen
- Caching Strategien
- Kompression

Schulungen & Termine

Link zur Termin Auswahl auf der Website:

workshops.de/seminare-schulungen-kurse/nestjs#alle-termine

Inhouse Schulung

- Termin nach Wunsch
- 3 ganze Tage oder 6 halbe Tage
- Remote oder vor Ort
- Deutsch und Englisch möglich

Kontaktiere uns für ein individuelles Angebot und frage deinen Wunschtermin an.

[Wunschtermin anfragen](#)

Kontakt

Öffnungszeiten: Werktags von 8 bis 13 Uhr

E-Mail: info@workshops.de

Telefon: +49 201 / 87535773