



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

