

22/01/2026

Tarun sharma

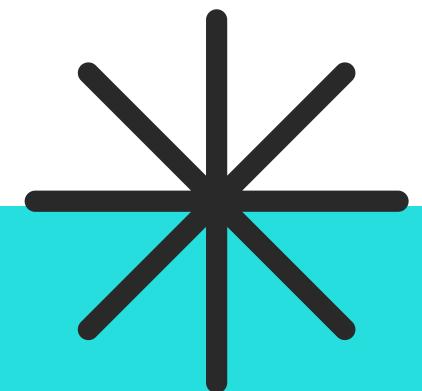
MASTERING GRAPHQL WITH NODE.JS & NESTJS

From Fundamentals to Apollo Federation



AGENDA

1. GraphQL Fundamentals & Core Concepts
2. Building & Securing GraphQL APIs
3. Advanced GraphQL with NestJS & Federation



GRAPHQL FUNDAMENTALS & CORE CONCEPTS



WELCOME TO GRAPHQL MASTERY

GraphQL: A Modern API Paradigm

Designed to address REST limitations and enable flexible data access.

Why GraphQL Matters Today

Empowers clients to request exactly what they need, reducing over-fetching.

From Basics to Federation

Journey covers core concepts, advanced patterns, and scalable architectures.

GRAPHQL CORE CONCEPTS

Schema-Driven API Design

GraphQL uses a strongly-typed schema to define API structure.

Types, Queries, Mutations, Subscriptions

Schemas include object types, queries for reading, mutations for writing, and subscriptions for real-time updates.

GraphQL vs REST: Conceptual Differences

GraphQL enables flexible, client-driven data fetching, while REST relies on fixed endpoints.



SCHEMA & TYPE SYSTEM DEEP DIVE

Scalar Types, Custom Scalars & Enums

Define basic data types, extend with custom scalars, and use enums for fixed sets.

Object, Input Types, Interfaces & Unions

Model complex data, accept structured inputs, and enable flexible type relationships.

Fragments & Field Nullability

Reuse query parts with fragments and control data reliability using nullability.



RESOLVERS & EXECUTION FLOW

Resolver Structure & Chaining

Resolvers map schema fields to data sources and can be chained for complex logic.

Key Resolver Parameters

Utilize parent, args, context, and info to access data and request details.

Field-Level Resolution

Each field can have its own resolver, enabling granular data fetching.

Error Handling Basics

Implement error handling within resolvers to ensure robust API responses.

BUILDING & SECURING GRAPHQL APIs

BUILDING WITH GRAPHQL YOGA

GraphQL Yoga Architecture

Lightweight, flexible server for rapid GraphQL development.

Schema-First Server Creation

Define your API using GraphQL SDL for clear, maintainable schemas.

Context Setup & Error Handling

Inject request-specific data and manage errors for robust APIs.



AUTHENTICATION & CONTEXT

Authentication in GraphQL APIs

Address stateless nature and unique security challenges

JWT-Based Authentication

Use JSON Web Tokens for secure user verification

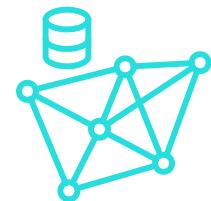
Context-Driven Authorization

Inject auth data into context for resolver access

Enforcing Access Control

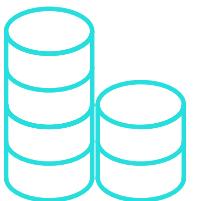
Apply authorization logic within resolvers

DATA FETCHING & PERFORMANCE



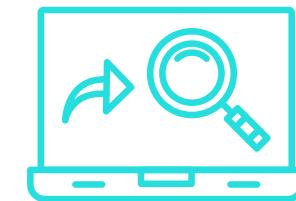
Efficient Database Integration

Connect GraphQL resolvers directly to data sources.



Solving the N+1 Query Problem

Use DataLoader to batch and cache database requests.



Implementing Pagination

Apply basic pagination for scalable data retrieval.

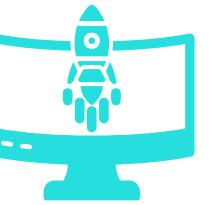
ADVANCED GRAPHQL WITH NESTJS & FEDERATION

GRAPHQL WITH NESTJS



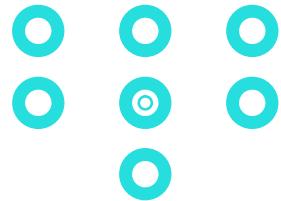
NestJS GraphQL Architecture

Leverages modular structure for scalable API development.



Apollo GraphQL Integration

Seamlessly connects Apollo Server with NestJS for robust GraphQL APIs.



Code-First vs Schema-First

Choose between TypeScript decorators or SDL for schema definition.

APOLLO FEDERATION ESSENTIALS

Why Apollo Federation Matters

- Enables scalable, modular GraphQL architecture

Supergraph & Subgraphs Explained

- Supergraph unifies multiple subgraphs into one API
- Subgraphs represent distinct services or domains

Federation Workflow Overview

- Compose schemas, route queries, resolve entities

Federation Directives & Entity Resolution

- Use @key, @extends, @provides, @requires for schema stitching
- Entity resolution links data across subgraphs

KEY TAKEAWAYS & NEXT STEPS

Master GraphQL Fundamentals

Grasp schema, types, queries, mutations, and subscriptions

Build for Production Readiness

Implement authentication, performance optimizations, and testing

Apply Advanced API Patterns

Utilize federation, context, and error handling for robust APIs

Accelerate Learning with Projects

Create hands-on APIs using Yoga, NestJS, and Apollo Federation