ONE PAGE SUMMARY

IT & Software › Backend Development › NestJS

Introduction to NestJS

Master Backend Development with NestJS Framework!

DEFINITION

> NestJS is a progressive Node.js framework for building efficient and scalable server-side applications. It uses TypeScript and is heavily inspired by Angular's architecture.

KEY CONCEPTS

Modules

Modules are the fundamental building blocks of a NestJS application, organizing related components.

Controllers

Controllers handle incoming requests and return responses, acting as the interface for client interactions.

Providers

Providers are classes that can be injected as dependencies, typically for services or repositories.

Middleware

Middleware functions are executed during the request-response cycle, allowing for request manipulation.

Dependency Injection

NestJS uses dependency injection to manage service instances, promoting modular and testable code.

EXAMPLES

→ Creating a simple REST API with NestJS and TypeScript.

→ Implementing middleware for logging requests in a NestJS application.

→ Using providers to manage database connections in NestJS.

MEMORY TIPS

★ Think of NestJS as Angular for the backend to remember its structure.

★ Use the acronym MCP (Modules, Controllers, Providers) to recall key components.

★ Visualize a tree structure where modules branch out into controllers and providers.

COMMON MISTAKES

✗ Neglecting to import modules in the main application module.

✗ Forgetting to register providers in the module where they are used.

✗ Confusing controllers with services in terms of their responsibilities.

QUICK RECAP

NestJS is a powerful framework for building backend applications using TypeScript. Key components include modules, controllers, and providers, which work together to create a structured application. Remember to leverage dependency injection for cleaner code.