TechZeen

Architecture Overview

- NestJS architecture is modular, scalable, and maintainable.
- This pattern helps in writing reusable and well-structured code.

</TechZeen

Client (User)

TechZeen

- The person or app that sends a request (e.g., from browser or mobile).
- Triggers the process by accessing an endpoint (like /products).

TechZeen

Controller

- Acts like a receptionist 🧑‍💼.
- Receives the client's HTTP request.
- Calls the right Service method to handle logic.
- Sends back a response to the client.

Service

- Contains business logic (e.g., fetch data, apply rules).
- Does not deal with HTTP directly—just logic.

TechZeen

Provider

- Any class that can be injected and reused (like Service, custom classes, etc.).
- Registered in the module to be used via Dependency Injection.

TechZeen

Module

TechZeen

- The container 📦 that groups Controllers, Services, and Providers.
- Organizes the app into features (e.g., ProductModule, UserModule).
- Helps keep the app scalable and clean.

Dependency Injection (DI)

- NestJS automatically provides services where they are needed.
- You don’t create new instances manually.
- Improves testability and reusability.

TechZeen

Decorators

TechZeen

- Special functions starting with @ (e.g., @Controller(), @Injectable()).
- Tell NestJS how to treat a class, method, or variable.
- Used for routing, injecting services, and more.