**🔁 FULL STEP-BY-STEP GUIDE: NestJS + PostgreSQL + TypeORM**

**✅ 1. Generate Module, Controller, and Service**

nest g module customer

nest g controller customer

nest g service customer

**✅ 2. Define DTO**

**Create DTO for input validation:**

// src/customer/dto/create-customer.dto.ts

export class CreateCustomerDto {

name: string;

age: number;

}

**✅ 3. Define Entity (Database Table)**

// src/customer/entities/customer.entity.ts

import { Entity, PrimaryGeneratedColumn, Column } from 'typeorm';

@Entity()

export class Customer {

@PrimaryGeneratedColumn()

id: number;

@Column()

name: string;

@Column()

age: number;

}

**✅ 4. Install and Configure TypeORM + PostgreSQL**

npm install @nestjs/typeorm typeorm pg

**Update app.module.ts:**

// src/app.module.ts

import { Module } from '@nestjs/common';

import { TypeOrmModule } from '@nestjs/typeorm';

import { CustomerModule } from './customer/customer.module';

import { Customer } from './customer/entities/customer.entity';

@Module({

imports: [

TypeOrmModule.forRoot({

type: 'postgres',

host: 'localhost',

port: 5432,

username: 'your\_db\_user',

password: 'your\_db\_pass',

database: 'your\_db\_name',

entities: [Customer],

synchronize: true, // Only in development!

}),

CustomerModule,

],

})

export class AppModule {}

**✅ 5. Import TypeORM in the Customer Module**

// src/customer/customer.module.ts

import { Module } from '@nestjs/common';

import { TypeOrmModule } from '@nestjs/typeorm';

import { CustomerService } from './customer.service';

import { CustomerController } from './customer.controller';

import { Customer } from './entities/customer.entity';

@Module({

imports: [TypeOrmModule.forFeature([Customer])],

controllers: [CustomerController],

providers: [CustomerService],

})

export class CustomerModule {}

**✅ 6. Implement Service Logic**

// src/customer/customer.service.ts

import { Injectable } from '@nestjs/common';

import { InjectRepository } from '@nestjs/typeorm';

import { Repository } from 'typeorm';

import { Customer } from './entities/customer.entity';

import { CreateCustomerDto } from './dto/create-customer.dto';

@Injectable()

export class CustomerService {

constructor(

@InjectRepository(Customer)

private customerRepository: Repository<Customer>,

) {}

async getAll(): Promise<Customer[]> {

return this.customerRepository.find();

}

async add(createDto: CreateCustomerDto): Promise<Customer> {

const customer = this.customerRepository.create(createDto);

return this.customerRepository.save(customer);

}

}

**✅ 7. Implement Controller Routes**

// src/customer/customer.controller.ts

import { Controller, Get, Post, Body } from '@nestjs/common';

import { CustomerService } from './customer.service';

import { CreateCustomerDto } from './dto/create-customer.dto';

@Controller('customer')

export class CustomerController {

constructor(private readonly customerService: CustomerService) {}

@Get()

getAll() {

return this.customerService.getAll();

}

@Post()

add(@Body() dto: CreateCustomerDto) {

return this.customerService.add(dto);

}

}