Certainly! NestJS is a powerful framework for building scalable and maintainable server-side applications. Below, I'll provide you with a basic example of setting up a simple backend application using NestJS with Node.js. Please make sure you have Node.js and npm installed before you begin.

1. **Install Nest CLI:**

npm install -g @nestjs/cliinstall -g @nestjs/cli

1. **Create a new NestJS project:**

nest new your-backend-app

cd your-backend-app

nest new your-backend-app cd your-backend-app

1. **Generate a controller:**

nest generate controller appnest generate controller app

1. **Edit src/app.controller.ts with a basic endpoint:**

**Generate a service:**

// src/app.controller.ts

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

@Controller()

export class AppController {

@Get()

getHello(): string {

return 'Hello World!';

}

}

nest generate service app

1. **Edit src/app.service.ts with a simple service method:**

nest generate service app

6. // src/app.service.ts

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

@Injectable()

export class AppService {

getHello(): string {

return 'Hello from the AppService!';

}

}

**7.Modify src/app.controller.ts to use the service:**

// src/app.controller.ts

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

import { AppService } from './app.service';

@Controller()

export class AppController {

constructor(private readonly appService: AppService) {}

@Get()

getHello(): string {

return this.appService.getHello();

}

}

This is a very basic setup. Depending on your application requirements, you can add more controllers, services, modules, and integrate with databases or external APIs. NestJS follows the modular architecture, making it easy to organize your code in a scalable way.

Top of Form