20 NestJS Interview Questions and Answers

Prepare for the types of questions you are likely to be asked when interviewing for a position where NestJS will be used.

[](https://climbtheladder.com/users/344/profile/interview-insights)

[Interview Insights](https://climbtheladder.com/users/344/profile/interview-insights)

Published Aug 12, 2022

NestJS is a server-side JavaScript framework that is becoming increasingly popular in the development community. With its ease of use and ability to create scalable applications, NestJS is a great framework to learn if you’re looking to enter the world of server-side JavaScript development. In this article, we’ll discuss some common questions you may encounter during a NestJS interview.

NestJS Interview Questions and Answers

Here are 20 commonly asked NestJS interview questions and answers to prepare you for your interview:

1. What is NestJS?

NestJS is a framework for building efficient, scalable Node.js server-side applications. It uses modern JavaScript, is built with TypeScript (preserves compatibility with pure JavaScript) and combines elements of Object-Oriented Programming, Functional Programming, and Functional Reactive Programming.

2. Who developed NestJS? Why did they develop NestJS?

NestJS was developed by Kamil Myśliwiec, who is a Polish software engineer. He developed NestJS because he wanted to create a framework that would be easy to use and would allow developers to create scalable server-side applications.

3. When was NestJS first released?

NestJS was first released on October 5, 2016.

4. What are some features of the NestJS framework?

Some features of the NestJS framework include its use of TypeScript, support for multiple database types, and its modular structure.

5. How do you install NestJS on your machine?

You can install NestJS by using the Node Package Manager (NPM). To do this, you will need to first install Node.js on your machine. Once you have Node.js installed, you can open up your terminal and type in the following command:

npm install -g @nestjs/cli

This will install the NestJS command line interface globally on your machine. From there, you can create a new NestJS project by running the following command:

nest new project-name

6. Can you explain what TypeScript is and how it relates to NestJS?

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript. It adds optional types to JavaScript that support tools for large-scale JavaScript applications for the first time. NestJS is a framework for building efficient, scalable Node.js server-side applications. It uses TypeScript as its programming language.

7. What’s the difference between Angular and NestJS?

Angular is a framework for building client-side applications, while NestJS is a framework for building server-side applications. NestJS is built on top of TypeScript and Express, and it aims to provide a more robust and scalable architecture for enterprise-level applications.

8. Is it possible to use other languages like Python or Ruby with NestJS? If yes, then how?

Yes, it is possible to use other languages with NestJS. NestJS is language agnostic, meaning that it can work with any language that can compile to JavaScript. This includes languages like Python and Ruby. To use a language like Python or Ruby with NestJS, you will need to use a language-specific compiler to compile your code to JavaScript.

9. What kind of applications can be built using NestJS?

NestJS is a framework for building server-side applications. It is built on top of TypeScript and Express, and it can be used to build a variety of different types of applications, from simple REST APIs to full-fledged web applications.

10. What are the main components of a NestJS application?

The main components of a NestJS application are the controller, service, and module. The controller is responsible for handling incoming requests and sending responses. The service is responsible for business logic and interacting with data sources. The module is responsible for organizing the application into cohesive units.

11. What are modules and why are they used in NestJS?

Modules are used in NestJS in order to group together related functionality and provide a higher-level structure for applications. By encapsulating related functionality into modules, NestJS applications can be more easily organized and maintained. This also makes it easier to reuse code across different parts of the application.

12. What are controllers and why are they used in NestJS?

Controllers are responsible for handling incoming requests and sending responses to the client. In NestJS, controllers are used to define routes and map incoming requests to the appropriate handler functions.

13. What are services and why are they used in NestJS?

Services are used in NestJS to encapsulate business logic and provide a way to share data and functionality between different parts of the application. Services can be injected into controllers and other services, making them a powerful tool for modularizing an application.

14. What does dependency injection mean in the context of programming? Why is it important for NestJS applications?

Dependency injection is a technique for decoupling the dependencies of a software component from the component itself. This means that the component can be used in different contexts without having to change its code. In the context of NestJS, dependency injection is important because it allows NestJS to provide the dependencies that a component needs at runtime, instead of having to hard-code them into the component. This makes NestJS applications more flexible and easier to change.

15. What is an interceptor in the context of NestJS?

Interceptors are functions that can be used to intercept incoming requests to a NestJS application and perform some sort of pre-processing or manipulation before the request is handled by the route handler. This can be useful for things like logging, authentication, or rate-limiting.

16. What are guards in the context of NestJS?

Guards are functions that can be used to run checks before a route is executed. For example, you might use a guard to check if a user is logged in before allowing them to access a route. Guards can be used to perform all sorts of checks, and they can be used in combination with each other to create complex security systems.

17. What are pipes in the context of NestJS?

Pipes are a feature of NestJS that allows for the transformation of data before it is passed to a route handler. This is useful for tasks such as validation or formatting data. Pipes can be chained together, and they can be used with middleware to create a powerful data processing pipeline.

18. What are middlewares in the context of NestJS?

Middlewares are functions that are executed by NestJS before a request reaches the final handler. Middlewares can be used for a variety of purposes, such as logging, authentication, and authorization.

19. What testing frameworks work best with NestJS?

NestJS is a Node.js framework, so any testing framework that works with Node.js will work with NestJS. Some popular options include Jest, Mocha, and Jasmine.

20. What databases work best with NestJS?

Any database that works with Node.js will work with NestJS. However, some databases are better suited for NestJS than others. For example, MongoDB is a popular choice for NestJS because it is a NoSQL database that is easy to use and scale.