Topic 5 Node.js and Express.js

Interview Questions and Answers

Q1. What is the package.json file in node.js?

Ans: Package.json plain JSON(JavaScript Object Notation) text file that contains all information/metadata about node.js application or project. Every node.js file should have this file in the root directory. Package.json contains the following metadata about the node.js project:

- **Name:** Name of the project
- **Version:** Version of the project
- **Description:** Description of the project.
- **Scripts:** Various scripts to run the application
- **Author:** Name of the author who developed the application.
- **Dependencies:** it contains the name and version of the package that is installed on your project/application.

Q2. What is express?

Ans:

- Express is a node.js framework that facilitates the management of data between servers and routes in the server-side web application.

- It is a lightweight flexible framework that provides a wide range of features for web applications.
- With help of express, you can easily build a different kinds of web applications in a short period.
- Express.js provides simple routing for requests and middleware that is responsible for making the decisions to give the correct response for the request made by the client.

Q3. What are the core modules of node.js?

Ans: Node.js provides some built-in modules which can be used without installing them on to your system. Following are some core/ built

- **HTTP:** it allows us to pass the data over hypertext transfer protocol. Using HTTP provides a method called createServer to create servers.
- **File System:** file system module allows us to perform various operations such as create, read, update and delete operations on the file system.
- **Operating System.** It provides information about computer operating systems.
- **URL module:** it is used various utilities for resolution and parsing. It helps us in splitting web addresses into various parts.

Q4. What are the different types of methods in node.js?

Ans: HTTP methods are used to perform various operations on databases. Following are some of the HTTP methods:

- **GET:** it is used to establish a communication and receive the data from the server. It Is used to make API calls when no modification is required on data.

- **POST:** post method is used to send through the body of the request. User login, file uploads, data uploads, etc. are done by post method.

- **DELETE:** This method is used to delete the data/resources from the database.

- **PUT:** This method is used to replace the existing resource on the database with updated resources.

- **DELETE:** It is used to delete a specified resource from the database.

- **PATCH:** it is used to perform modification on the database.

Q5. What is middleware?

Ans: Middleware functions are functions that have access to the request object (req), the response object (res), and the next function in the application's request-response cycle. The next function is a function in the Express router that, when invoked, executes the middleware succeeding the current middleware.

- Middleware functions can perform the following tasks:
- Execute any code.
- Make changes to the request and the response objects.
- End the request-response cycle.
- Call the next middleware in the stack.

Q6. What are the problems before node.js?

Ans:

- Fetching of data from database is time-consuming.
- Multithreads were used before node.js which results in deadlock and race conditions.
- The thread which is used for one request cannot be used for another request at a given point of time, so each time a new thread has to be assigned to the request, but if all threads are busy, then new request has to wait till old threads gets free.
- Processing of more requests takes more hardware which is costing more.

Q7. What are status codes?

Ans:

- **Informational status code(1xx)** – request has been received and the process is continuing.
- **Successful status code(2xx)** – response has been returned.
- **Redirects status code(3xx)** – further action must be taken in order to complete the request.
- **Client error status code(4xx)** – request contains incorrect syntax or cannot be fulfilled.
- **Server error status code(5xx)** – server failed to fulfil a valid request.

Q8. What is a Heap?

Ans: Heap is a binary tree data structure where the root node is compared with its children. It consists of two types:

1) **max heap:** a max heap is a complete binary tree in which value of each is greater than or equal to its child nodes

2) **min heap:** a max heap is a complete binary tree in which value of each is less than or equal to its child nodes

Q9. What is the difference between global packages and local packages?

Following are the major differences between global and local packages

Local packages	Global packages
It is installed in the directory where you run the **npm install <package name>**, by placing their codes under node_modules	It is placed in a single place of your system, regardless of where you run **npm install -g <package-name>**
It is imported to your code by **require('package-name')**	It is imported to your code by **require('pacakge-name')**
Can have different versions of the same package in different application	Can have only a single version of the same package
Easy to handle when there is an upgradation of a package	Difficult to handle when there is an upgradation of package
Occupies more memory	Occupies less memory
Use it to execute any type of command	Use it execute shell commands

Q10. What is the difference between Relational and NoSQL DBMS?

Features	Relational DBMS	NoSQL DBMS
Schema	It is necessary to define rows and columns before storing the data	It is schema-less and drops data into documents
Representation	Represents data in rows and columns	Represent data as a collection of JSON documents
Scaling	Does not scale easily	Scaling is automatic and transparent
Query	It uses Structured Query language.	NoSQL DBMS like MongoDB uses object querying.
Application	It is a good option – • If data structure fits nicely into tables • If requires SQL • In case of transactions	It is a good option • If data seems complex in relational database system • if pre-defined schema is not required • if you want to store records

		in the same that have different fields
performance	Poor performance when using Object Relational Mapping	Perform much better when using object relational mapping

Q11. What is a Database?

Ans: Database is an organized collection of data or information so that it can be easily accessed, updated, or manipulated.

- Database management system is a software that is used to define, update, retrieve and manage data in database
- The majorly used DBMS are relational database management and NoSQL database management system
- Example of some of database management software are Microsoft access, MySQL, Oracle, MongoDB, etc.

Q12. What is CORS?

Ans:

- CORS stands for cross origin resource sharing; it is a http header-based mechanism.
- It allows a server to indicate any origins such as domain, scheme, or port other than it's own from which browser should permit loading resources.
- It relies on a browser mechanism to make a preflight request to the server hosting the cross- origin resource, in order to check that server will permit the actual request.
- Preflight request is a small request sent by browser before actual request.
- It handles important information like which type of HTTP method will be used, also whether any custom HTTP header is present in it.

Q13. What is MVC architecture?

Ans: Model-View-Controller is an architectural pattern commonly used for developing user interfaces that divides the application into interconnected parts, Model, View and Controller.

- **View:** this layer focuses on the client side. The goal this view is, it represents the data on the data using templates, layouts, etc. it should not have knowledge of business logic or couple of the logic.
- **Controller:** this is the middle layer that takes the requests from the view and interacts with backend (Model) and renders the details back into view accordingly.
- **Model:** This is the backend layer, which focuses on logic which focuses on business logic or database interaction. This will interface with the

controller to receive the data from the view and sends the response back to the view