31. Difference between spawn() and fork() methods in Node.js?

spawn() - is used to launch a new process with the provided set of commands

fork() is a special instance of spawn() that executes a new instance of the V8 engine

**32. What do you understand by global objects in Node.js?**

Global are the objects which are global in nature and are available in all the modules of the application.

You can use these objects directly in your application, rather than having to include them explicitly

The Global objects can be modules, functions, strings, object, etc

Some of these objects can be in the module scope instead of Global Scope.

**33. Explain the concept of stub in Node.js**

Stubs are basically the programs or functions that are used for stimulating the module or component behavior

During any test cases, stubs provide the canned answers of the functions

**34. How assert works in Node.js?**

Assert is used to write tests

It only provided feedback only when any of the running test cases fails

This module gives you a set of assertion tests which are then used for testing invariants

It is basically used internally by Node.js but using require('assert') code, it can be used in other applications as well

**35. Define the concept of the test pyramid. Explain the process to implement them in terms of HTTP API**

The test pyramid is basically a concept that is developed by Mike Cohn

You should have a higher number of low-level unit tests as compared to high-level end-to-end tests that running through a GUI.

In terms of HTTP APIs it may be defined as:

A higher number of low-level unit tests for each model

Lesser integration tests to test model interactions

Lesser acceptance test for testing actual HTTP endpoints

**36. Explain the purpose of Express.js package?**

Express.js is a framework built on top of Node.js that facilitates the management of the flow of data between server and routes in the server-side applications.

It is a lightweight and flexible framework that provides a wide range of features for web applications

Express.js is developed on the middleware module of Node.js called connect

If you are working with any of the connect based middleware modules, then you can easily integrate with Express.js

**37. Differentiate between process.nextTick() and setImmediate() ?**

process.nextTick() and setImmediate(), both are functions of the Timers module which help in executing the code after a predefined period

**38. Explain the usage of a buffer class in Node.js?**

Buffer class in Node.js i used for storing the raw data in a similar manner of an array of integers

But it corresponds to a raw memory allocation that is located outside the V8 heap

It is a global class that is easily accessible can be accessed in an application without importing a buffer module.

Buffer class is used because pure JavaScript is not compatible with binary data

**39. How does Node.js handle the child threads?**

Node.js is a single threaded process and does not expose the child threads or thread management methods.

You can still make use of the child threads using spawn() for some specific asynchronous I/O tasks

If you still want to use the threading concept in your application you have to include a module called ChildProcess explicitly.

**40. Explain stream in Node.js along with its various types.**

Streams in Node.js are the collection of data similar to arrays and strings. They are objects using which you can read data from a source or write data to a destination in a continuous manner

These streams are expecially useful for reading and processing a large set of data.

In Node.js, there are 4 fundamental types of streams:

1. Readable

2. Writable

3. Duplex

4. Transform

**41. What is the use of Node\_ENV?**

If the project is in the production stage, Node.js promotes the convention of making use of Node\_ENV variable to flag it

This helps in taking better judgment during the development of the projects

Setting your Node\_ENV to production, makes your application to perform 3 times faster

**42. Difference between readFile vs createReadStream in Node.js?**

**43. List down the various timing features of Node.js?**

**44. Explain the concept of Punycode in Node.js?**

punycode = require('punycode');

**46. Does Node.js provide any Debugger?**

Node.js do provide a simple TCP based protocol and debugging client that comes built-in.

node debug [script.js | e "script" | <host> " <port> ]

**47. Describe the exit codes of Node.js**

**48. What is Event Emitter in Node.js**

Event Emitter is a Node.js class that includes all the objects that are capable of emitting events.

**49. Is cryptography supported in Node.js**

const crypto = require('crypto');

**50. Why Express 'app' and 'server' must be kept separate.**

Faster testing execution

It allows testing the API in-process without having to perform the network call

Better separation of concerns and cleaner code