1)token?

In jwt token once we are login server is recognised me and send me the whole details .

And I working in my account .and server recognised me until i am not logout the application

In this contain

Header=>which algorithm, type,

Payload=> data, time for token creation,

Signature:secret key

Jwt.sign(payload, signature, { expiresIn**:** 60 **\*** 60 });

Jwt.verify (token , signature).

Note: we are not pass the important information on payload we just pass user id.

2)node ?

3)node emitter ?

4)parallel execution in nodejs?

Means node js perform asynchronous opetaion

7) joins in mongodb?

8)fork?

9) node js

10) What is event listener?

**Event Listeners** are similar to call back functions but are associated with some event. For example when a server listens to http request on a given port a event will be generated and to specify http server has received and will invoke corresponding event listener. Basically, Event listener's are also call backs for a corresponding event.

Node.js has built in event's and built in event listeners. Node.js also provides functionality to create Custom events and Custom Event listeners.

11)**node js is single threaded and how they handle the concurrency?**

Node provides a single thread to programmers so that code can be written easily and without bottleneck. Node internally uses multiple POSIX threads for various I/O operations such as File, DNS, Network calls etc.

When Node gets I/O request it creates or uses a thread to perform that I/O operation and once the operation is done, it pushes the result to the event queue. On each such event, event loop runs and checks the queue and if the execution stack of Node is empty then it adds the queue result to execution stack.

12) **What is an error-first callback?**

Error-first callbacks are used to pass errors and data. The first argument is always an error object that the programmer has to check if something went wrong. Additional arguments are used to pass data.

13)Node Forever?

1. **Forever** is **Process Manager** for **Node.js applications**.
2. Forever keeps node.js applications running on production server.
3. Forever automatically restart server if it crashes or close, without any downtime.

14)Node PM2?

1. **PM2** is **Process Manager** for **Node.js applications**.
2. PM2 has built-in **Load Balancer**.
3. PM2 keeps node.js applications running on production server.
4. PM2 automatically restart server if it crashes or close, without any downtime.
5. PM2 helps you to manage application logging, monitoring, and clustering.

15. Middleware?

app.use((err, req, res, next) => {

if (! err) {

return next();

}

res.status(500);

res.send('500: Internal server error');

});

16. Event Emitter?

Node.js a built-in module called "events".

* In Node.js, EventEmitter is used to "emit" an errors.
* In Node.js please ensure that if any event emitters emit an error that event should be listened properly, otherwise your app will crash.
* const EventEmitter = require('events');
* //Creating object of EventEmitter class
* const emitter = new EventEmitter();
* // Listening the event
* emitter.on('logs', function() {
* console.log('Hello, welcome to EventEmitter in Node.js!')
* })
* // Creating an event
* emitter.emit('logs');

## 17. Error first callback - Node.js?

function(err, data) {

## 18. Node.js Inspector Module?

## Debug the node application

## 19. How do I exit/end a node.js process?

## Process.exit()

## Press ctrl +c

## 20. How to resolve cross origin (CORS) issue in node.js?

**There are mainly three ways you can do this using**

1. res.setHeader()
2. res.header() OR res.set()
3. express cors module

21. **How to catch all uncaughtException for Node.js?**

Use process **'uncaughtException'** and **'unhandledRejection'** events

## process .on('unhandledRejection', (reason, p) => { console.error(reason, 'Unhandled Rejection at Promise', p); }) .on('uncaughtException', err => { console.error(err, 'Uncaught Exception thrown'); process.exit(1); });

## 22. What is package.json in node.js?

1. The package.json is a json file which holds all the **metadata information** about your node.js Application.
2. NPM uses this **package.json** to manage all modules/packages dependencies for your node.js application.

**23. Q:- Difference between package.json and package-lock.json?**

**package.json:**The package.json is used for more than dependencies - like defining project properties, description, author & license info etc.

**package-lock.json:**The package-lock.json is primarily used to lock dependencies to a specific version number.

**24. Q:- Explain how do we decide, when to use Node.js and when not to use it?**

**Use Node.js:**

* Single Page Applications
* Real-Time services (Chat Applications, Games Servers etc)
* REST APIs and Backend Applications
* Blogs, CMS, Social Applications.
* Data Streaming Applications
* Utilities and Tools
* Advertisement Servers.

**Don't use Node.js:**

we should not use node.js for cases where the application requires long processing time. So, Node.js is best suited when processing needs less dedicated CPU time.

25) pipes in node js?

<https://www.fullstacktutorials.com/interviews/node-js-interview-questions-29.html>

25) if we logout the functionality which method wewant?

Put method we want