Candidates or Interviewer can submit questions they were asked or would like their candidates to know and probable answers to them

Node.js-Interview-Questions

Candidates or Interviewer can submit questions they were asked or would like their candidates to know and probable answers to them

1. How is Node.js different different from Javascript?
   Ans: Node.js uses a library called libuv, to bring an asynchronous event driven model to Javscript. It piggy backs on Chrome's Javascript Engine called V8 to enable the use of Javascript's call stack.

2. What is “callback hell” and how can it be avoided?
   Ans: Callback hell is when you have a lot of callbacks inside callbacks. Which keeps indenting the code to the right, to a point where it is no longer readable and nor maintainable. There are two ways to avoid it. i. By the use of a promgramming construct called **Promises**. Using promises you could still write callbacks, but now they were chainable. So instead of code indenting to right. It now increased in vertical direction. Which solved the major problem of callback hell, but still the code was verbose and lot of thens had to be written which made it a little less pleasing.
   ii. Using the new **async/await**. Async await removes the .then blocks of your code and replace them with try and catch blocks which are usually the traits of a synchronous design pattern.

3. When Should We Use Node.Js?
   Ans: We should use Node.js when we are building systems which require high number of I/O operations. Like saving data to DB, files access. NodeJS shines in such situation. For Ex : IoT Applications, Video Streaming.

4. When Node.js should not be used?
   Ans: Node.js should not be used when you have lot of synchronous code that needs to be run. Applications which require number crunching or data analysis should not use Node.js. For Ex. Matrix Multiplication, Summation, Aggregation of large datasets. In such cases, applications which allow Multi Threading out of the box like Python, Java should be used.

5. Name 3 ways to manage events in an asynchronous way?
   Ans: First, a classic use of callback methods. Second, using Promises. Finally, in a more sophisticated way, we can use reactive extensions (RxJS) to define pipelines the manage events asynchronously

6. What are Promises?

7. What is the difference between Node.js vs Ajax?
8. Is Node.js really Single-Threaded?
9. Can you explain what is Globals in Node.js?
10. What is the Use of underscore in Node.js?
11. What is Event Loop and Event Emitter?
12. Why to use Buffers instead of binary strings to handle binary data ?

Answers