Humza Hasan

WHAT IS node JS™

- (!programming language)
- Javascript runtime enviroment
- Built on Chrome V8 engine
- JS Code on Server Side

Thoda Gyaan……

- Node.JS is a way to run JS code on the server side as oppose to be the original JS which only worked on client side ! Node.js is JS runtime built on Chrome's V8 engine !
- Why Chrome V8 ? written in C++ thus making it very fast in compilation because its fastest.
- Node.JS is not a programming language. Its runtime JS code ! Both Chrome and Node.JS are written in C++ ! which runs on V8 engine which is also written in C++.

WHY USE NODE.JS ?

- Event Driven
- Non Blocking
- Asynchronous
- Single Thread

- **Blocking Example :**

```
// assume getData to be an API call
console.log("Getting Data1");
var data1 = getData('123');
console.log("Data is:", data1);

console.log("Getting Data2");
var data2 = getData('456');
console.log("Data is:", data2);

var sum = 1 + 2;
console.log("sum is:", sum);
```

Output :

```
Getting Data1
Data is: {“id” : 123, “name” : “chennai”}
Getting Data2
Data is: {“id” : 456, “name” : “mumbai”}
sum is:3
```

- **Non - blocking Example :**

```
console.log("Getting Data1");
getData('123', function(data1) {
  console.log("Data is:", data1);
});
console.log("Getting Data2");
getData('456', function(data1) {
  console.log("Data is:", data1);
});

var sum = 1 + 2;
console.log("sum is:", sum);
```

Output :

```
Getting Data1
Getting Data2
sum is:3
Data is: {“id” : 123, “name” :
“Captain America”}
Data is: {“id” : 456, “name” : “Iron
Man”}
```

Bird’s Eye View

- The non-blocking form will get executed faster than the blocking form and hence code written in blocking form (in other languages) will take time than the one written above, hence the power of Nodejs non-blocking I/O mode kicks in.

Node.JS has an amazing feature of being of Asynchronous nature

Synchronous :

```
console.log("SYNC Program Starts");
console.log("DB value fetched !!");
console.log("SYNC Program Ends");
```

OUTPUT :

```
SYNC Program Starts
DB value fetched !!
SYNC Program Ends
```

Asynchronous :

```
console.log("SYNC Program Starts");

setTimeout(function(){
    console.log("DB value fetched !!");
},2000)

console.log("SYNC Program Ends");
```

OUTPUT:

```
SYNC Program Starts
SYNC Program Ends
DB value fetched !!
```

Now comes the funny part….

- Node.JS is single threaded, why ?
- Because it is built on Chrome V8 engine which is also runs on single thread architecture
- And why is that ? Because with lots of trial and error, hit and run , shoot and recover we found that single thread gives much better performance than one thread per request.

Modules in Node.JS

- 1. **Core Modules**

- – Global , eg : console.log() , fs, http
- – Installed Modules , eg : express , mongoose

- 2. **Third Party Modules**

```
const validator = require('validator');
console.log(validator.isEmail('humza1mza.co'));   //false
console.log(validator.isURL('http//www.npmjs.com/package/validator'));    //false
console.log(validator.isDivisibleBy('1234',4));     //false


const chalk = require('chalk');
console.log(chalk.green("Success !!!"));
console.log(chalk.bold.underline.bgRed("Warning !"));
```

3. User Created Modules

<u>App.js</u>

```
const name = "Humza";
const util = require('./util.js');
console.log(name);
console.log(util.loca);
console.log(util.add(5,5));
```

<u>Util.js</u>

```
console.log("Util Package");
var loca = "Kolkata";
const add = function(a , b) {
    return a+b;
}
module.exports = {
    add,
    loca
}
```

DEbugging Node.JS

- console.log()

- Debugger statement before the code you want to check, which pauses the flow of the program and you can dry-run the code in chrome://inspect which opens us the chrome developer tools

Any Questions ?