\*\*\*\*\*\*\*\*\*\*Search answers in Node\_Answers doc also.

==============Accenture Interview Starts=========================

Resume related

What is clouser

What is callback

What is callbackhell ? how to avoid

Threadpool in node?

Difference between setTimeout(fn, 0)and **setImmediate**

**Mention the framework most commonly used in node.js?**

“Express” is the most common framework used in node.js

**What is express , loopback**

**Eventloop**

**What is use of app.use**

**Middlewares**

SSL layer

How you do unit testing of your service

How you do integration testing …(Supertest??)

mongoDB connector name, how you connected to mongoDB

AWS, DOM

==============Accenture Interview Ends=========================

================Capita Interview (Siddhi) Starts=====================

explain which node modules you are using in your API and use of each(list express , body-parser, dbconnector models as a priority)

sum(2,3) and sum(2)(3) should go to same function definition.

sum(2)(3) and sum(2)(3)(4)(6)……………….any no of parameter…. should go to same function definition.

Variable hoisting

Splice method

Second max number from array

================Capita Interview (Siddhi) Ends=====================

Built-in access-level controls

Auth

================== Neo-soft Interview (Sandip) Starts=====================

Ng-app can be used more than once in one module?(anular js question)

What is piping

What are bouncy Script ( - Used to provide port dynamically at run time)

How to create server in express

How to create server without express (using http.createServer)

What is REST – Diff bet SOAP and REST ? Why we go for REST

What is JSON

What is Parser

AWS

What are Streams

How you avoid callbackHell in your API(Using Async Apis)

What is prototype

Use of .exports();

================== Neo-soft Interview (Sandip) Ends=====================

==============Hexaware online test - Sandip Starts=========================

Each 10 questions

**1 . JavaScript**

Strings methods related

Prototype

Inheritance

Real number

arrays

**2. Node**

ReadStream Write Stream , fs. … Syntax

HTTP questions

Package. Json contents what?

how to install a node module outside your working directory – npm install module –g

**3. Restful webservices- JAX-RS..**

Annoatations,

If you want to send status code along with object or String :

Ans: Response.status(200).entity(Object OR String).build();

Init-param use in web.xml

How to add header to response

How to use two types of data types in @Produces:

Ans : @Produces({"application/xml", "application/json"})

You can pass parameters to a servlet from the web.xml file. The init parameters of a servlet can only be accessed by that servlet. Here is how you configure them in the web.xml file:

<servlet>

<servlet-name>controlServlet</servlet-name>

<servlet-class>com.jenkov.butterfly.ControlServlet</servlet-class>

**<init-param>**

**<param-name>myParam</param-name>**

**<param-value>paramValue</param-value>**

**</init-param>**

</servlet>

@produses ==== @consumes

Intellect Design:

Elasic Search,

Main deff bet SQL and NoSQL

What is the need of use of streams

Mocha chai gulp

Node modules used in project

Infosys:

Siddhi- mongoos,

Degugging in node (plugin)

Git commands – from checkout to update

SDLC implementation

Socket programing

Websocket

Routing related

And to end login module design ..get data from front end ..middleware and update it into DB

Shubham- mongoos,

Degugging in node (plugin)

Login module code- from front end to banckend

Git commands – merge req craetion

Angular 2

Node and anugale version – working and the latest

Callback hell, how to avoid, how modularization is done

What is ajile

ACMA6 me generator kya karata hai

Single threred

Asynchronous

Promises - writing

--save

==============Build IO =======================

How you authenticate your API

How you generate token

When it expires

Clustering in Node Js

Socket programming

Closure

Closure real time example

Node Architecture - How everything works in Node.

What is Event loop

Are you aware about ES6 concepts

Node modules used

Mocha Chai (do you use mock testing ?)

Difference between put and post

Arrow function

Package.json parameters

What is Child process

**Mongo –**

What are Operators ?

Query :- get only Name column for all users from USER collection .

========================Cognizant ========================

Challenges faced in your API

Current version of JavaScript and node and which you are using

Are you using promises

What is promises and callback

Asynch coding?

Node modules used

Why not using ES6

What is strongloop

Difference between postman and restClient

Answers: =================================

Difference between setTimeout(fn, 0)and **setImmediate**

**setTimeout(fn, delay)** calls the given callback fn after the given delay has ellapsed (in milliseconds). However, the callback is not executed immediately at this time, but added to the function queue so that it is executed **as soon as possible**, after all the currently executing and currently queued event handlers have completed. Setting the delay to 0 adds the callback to the queue immediately so that it is executed as soon as all currently-queued functions are finished.

**setImmediate(fn)** achieves the same effect, except that it doesn’t use the queue of functions. **Instead, it checks the queue of I/O event handlers**. If all I/O events in the current snapshot are processed, it executes the callback.

console.log("first");

**setTimeout**(function() {

console.log("second");

}, 0);

console.log("third");

The output will be:

first

third

second

console.log("first");

**setImmediate**(function(){

console.log("second");

});

console.log("third");

The output will be:

first

third

second

console.log("first");

**setTimeout**(function() {

console.log("secondTime");

}, 0);

**setImmediate**(function(){

console.log("secondImmediate");

});

console.log("third");

The output will be:

first

third

second**Immediate**

second**Time**

**What will be the output of bellow:**

**for(var i=0;i<5;i++){**

**setTimeout(function() {**

**console.log("value is "+i);**

**}, i\*1000);**

**}**

**Output**

value is 5

value is 5

value is 5

value is 5

value is 5

**What is Callback Hell and how to avoid it ?**

Callback hell means a heavily nested callbacks which make the code unreadable and difficult to maintain. To avoid callback hell one should use **modularization**, which means breaking the callbacks into independent functions. Another method to avoid callback hell is use to use **Promises**, which allows error propagation and chaining.

**REPL**

**And its commands**

### List And Explain The Important REPL Commands?

**Answer.**

Following is the list of some of the most commonly used REPL commands.

* **<.help> –** It displays help for all the commands.
* **<tab Keys> –** It displays the list of all the available commands.
* **<Up/Down Keys> –** Its use is to determine what command was executed in REPL previously.
* **<.save filename> –** Save the current REPL session to a file.
* **<.load filename> –** To Load the specified file in the current REPL session.
* **<ctrl + c> –** used to Terminate the current command.
* **<ctrl + c (twice)> –** To Exit from the REPL.
* **<ctrl + d> –** This command perfoms Exit from the REPL.
* **<.break> –** It leads Exitting from multiline expression.
* **<.clear> –** Exit from multiline expression.

What is Clouser

<http://javascriptissexy.com/understand-javascript-closures-with-ease/>

==========================Neo-Soft 1=====================

banking domain project ...wht you do for security

where exactly do authentication

let var diff

let const diff

object defind with const—and trying to add one more parameter to and object ..will it work

what you do for error handling

where exactly you are doing error handling

what are middlewares in nodeJs..importance

how to write middleware at route level

scenario – API 1 and API2 -- …you want to use one middleware for API1 and not API2 --how

middleware custom use

Application level middle-ware use

----

use same authentication logic for api1 and api2 -- how -- Application level middle-ware use

---

API level middle-ware use

session and cookie difference

add cookie to header

how to restrict the modification of cookie

which logging module is used to log

when to use node

What is the js function keyword used to chk the datatype of variable

How to differentiate the exact datatype of object and array

-----------

action 1

action 2

do something after both

eval function

arrow function

which db using

how you run multiple request (multiple independent functions) parallely

how to identify code is running in prod or dev

when to use clouser

node advantages

cpu intensive apps ex

how to decide we should go for node

accept and content type

es6 concepts

challenges faced?

How node js works

Npm and yarns.

Console statement .. print on both browser and node ….which will print fast.==(ANS node I think)

Whole concept of closure

Promises

Diff bet undefined and not defined

Diff bet null and undefined

How to make variable and null and undefined

In packg.josn how to give exact dependency

Restrict the version of dependency in pck.json

How to use a module in development mode

Have you set up the code on severer

how to run the code in production env? What is command

how to run code in local

What is node pm2?

**PM2** is a production process manager for **Node**.js applications with a built-in load balancer. It allows you to keep applications alive forever, to reload them without downtime and to facilitate common system admin tasks.

What are node environment variable

============================NeoSoft f2f==============

What is event emitter.. why do we use it

File system

How do you manage files in your project

How to truncate file

How to upload file from front end to sever and store it in db

Which db you

Ways of API security

Passport js

Diff bet no sql and rdbms

Diff bet pm2 and ..??

Explain whole project and framework you use

Singleton

=================deloitte =========================

IIFE

Closuer

Typeof

Option method

========================ZYMER pune ===================

How to increase API security

How you are doing Authentication

What is child process

How to communicate between one child node to another

How to start child process

How to scale up application

What is Pipline why to use it

http framework

gulp

JSHint?

Promises …no need to require from v4

This keyword use

How to do error handling in node

=== ZYMER round 2 ===

In java script is possible to have 2 different function with same name?

What is stub

How to ensure that func2 has been completed after func1

What is event bubbling

what is anonymous function

what is event loop

what is closure

what are event emitter

error first callback

what is strict mode

what are global in node js

the function can be published with the different name?

================== Synecron 1 and 2 ============

Design patterns in node Js

How to create class in node

Create objet using construct pattern

Imp of streams

Imp of events

Imp of fs

How to deploy code on server

Prototype

Why DOM can’t be accessed in node

Diff be foreach and .map

Context of THIS

Call apply and bind

Diff bet strongloop and express .. which one mvs (ans - strngloop)

Node and express version

.pipe syntax

Middleware

Arrow function advantage

Clustering

Web socket

Istanbul

How you generate code coverage report

When to use node js

Event emitter

Where you store the access token

Get and POST are default methods for web browser .. how to go for others

Put can also insert new data, then how to diff bet put and post

How to send runtime args to node js..(process.argv??)

================== Synecron 3 ============

JSON validator , how you use in project

.map and forEach difference

For and forEach difference

Yarn and npm diff

In promises whento use reject and when to use throw err ..can we use thero err in promises

How you mock database connection for Unit testing

All \* plugin …

What if I use arrow function as a constructer?

What is test pyramid ?

Object.keys

<https://scotch.io/tutorials/javascript-promises-for-dummies>

Chaining of promises ex

// Promise

var isMomHappy = true;

// 1 st promise

var willIGetNewPhone = new Promise(

function (resolve, reject) {

if (isMomHappy) {

resolve('From 1st Promise resolve '); // fulfilled

} else {

var reason = new Error('mom is not happy');

reject('1st reject '); // reject

}

}

);

// 2nd promise

var showOff = function (phone) {

return new Promise(

function (resolve, reject) {

var message = 'From 2nd promise ' ;

resolve(' 2nd resolve');

//reject('2nd Rject ');

}

);

};

// call our promise

var askMom = function () {

willIGetNewPhone

.then(showOff) // chain it here

.then(function (fulfilled) {

console.log("final resolve "+fulfilled);

// output: 'Hey friend, I have a new black Samsung phone.'

})

.catch(function (error) {

// oops, mom don't buy it

console.log("Fianl Rejact "+error);

// output: 'mom is not happy'

});

console.log('after asking mom');

};

askMom();