Node js:

1. Concepts:

* Node.js is a server side framework.
* Node js is get multiple request and create thread for that each request. If request thread is finish work then it will run and notify to the node. The node check the notifications and then give response to the user.
* Concepts of node js are:
* Immediately invoked function expressions:
* IIFE is a function that’s execute when it’s create.
* IIFE function definition:

(function(){

//coding….

})();

* Closures:

A closure in JavaScript is an inner function that has access to its outer function's scope. A closure makes the variables of the inner function private.

* Prototype:

A prototype property that is used to attach properties and methods. JavaScript supports inheritance only through the prototype property.

* Method overloading:

Method overloading allows multiple methods to have the same name but different arguments. The compiler or interpreter determines which function to call based on the number of arguments passed.

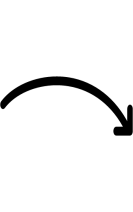
* Bind method:

The bind method allows you to pass arguments to a function without invoking it.

2. Event loop:

* Node js is a single-threaded application.
* Event loop control the request and responses through callback.
* Each and every request need to perform different task. If task is finish then it notify to the node js.
* The node js check the operation is finish or not. If it is done then it will send that details back to the user.

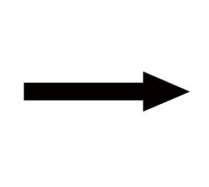
Register callback



Operations

File system

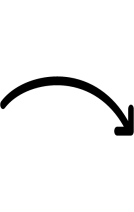
Requests(multiple)

 Non-blocking

Database

computation

Trigger callback I/O operation.



operation

complete

3. Loggers (winston):

Logger Means:

* The activity or business of felling or cutting trees and preparing the timber.

Basic-logger:

* Basic logger for nodejs supporting error, warning, information, debug and trace messages with (or without) timestamp. Everything you log is printed to the console.
* The logger is used to display the error, warning and debugs etc.

Four logger libraries are:

1. Node-loggly
2. Bunyan
3. Winston
4. Morgan.

4. Node Modules (npm):

1. Module in Node.js is a simple or complex functionality organized in single or multiple JavaScript files which can be reused throughout the Node.js application. Each module in Node.js has its own context, so it cannot interfere with other modules.
2. To include module, use the require() function with the name of the module.

Ex: var http=require(‘http’);

5. Express JS:

* Express - the most popular framework for Node.js
* It adds support for routing, middleware, view system and many more.
* It is an open source framework developed and maintained by the Node.js foundation
* Express is quite simple and easy to use, unlike other frameworks that reduce the flexibility to have design choices by trying to do way too much than expected.
* Express.js is a framework based on Node.js for building web Application using principles and approaches of Node.js.
* Express example:

var express=require(‘express’);

var app=express();

app.get(‘/’,function(req,res){

res.send(“welcome”);

});

app.listen(3000);

output:

welcome.

6. Mongoose:

* Mongoose is a JavaScript framework that is commonly used in a Node.js application with a MongoDB database.
* Mongoose is an Object Document Mapper (ODM). This means that Mongoose allows you to define objects with a strongly-typed schema.
* Mongoose currently contains Eight SchemaTypes, that are:
  + - * String
      * Number
      * Date
      * Buffer
      * Boolean
      * Mixed
      * ObjectId
      * Array

7. Basic code structure:

1. package.json

2. server.js

3. database[used to add or retrieve the data]

4. model

5.source

* Resource

Implementation

* Service

6.tests

7.confiqurations

8.routes

8. Using 'exports.functionName':

* Export is used to access the method which is not in that module file. That is export keyword make methods available outside the module file.
* Example:

File1.js

Var http=require(‘http’);

Var date=require(‘./file2.js’);

http.createServer(function(req,res){

res.write(“date and time is:”+date.myDateTime());

res.end();

}).listen(8080);

--------------------------------------------

File2.js

exports.myDateTime=function(){

return Date();

};

* Here we find date and send that details from one file to another file using exports keyword.