Introduction to Node.js

==>Earlier js was only meant to be executed on the browser. Javascript code runs on browser only, we cannot run js code outside the browser.js engine is responsible for js execution but js engine is available inside browser

==>why not to use js on a server side?

==>developer of node js is RYAN DAHL -->embed v8 engine of chrome brower with c++---------->node js

==>C++ is considered a native language because it compiles directly into machine code that can be understood by the underlying system

==>using node js we can directly run/executes .js file and results will be display on the nodejs console.

1)we can run js outside of the browser

2)js can talk to native m/c because of c++ .js code is now executed using v8 engine on c++

i.e we can use js for desktop,web application.

3)if we add http component inside node js,we can run node js on server side also.it is server side technology that run on v8 compile(chrome engine)

==>node js is an application for executing javascript code. runtime environment for js.

==>node js is not an language or library

====>node js is a runtime env for executing js code

==>uses async tech(concurrently executing task)

Node.js is a cross-platform runtime environment for running JavaScript applications outside the browser. It is used for creating server-side and networking web applications.

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications.

Node.js also provides a rich library of various JavaScript modules to simplify the development of web applications.

Node.js = Runtime Environment + JavaScript Library

===================================================

download nodejs

https://nodejs.org

====================================================

command to execute js code using nodejs:

node filename.js

===============================================================

module:

==>it is a javascript file which is a collection of various function,classes and properties.

==>reusability

Types of modules

1)Built in modules==>when u install node js in your system at that time we will get built in modules,already builtin modules in js

ex:http(to create our own server)(we want to create javascript program that should become capable of listing to a request,processing ,return a response),for that we shld get predefined functions in a module i.e http

ex:fs

ex:url

==>c,c++ we use to including headerfiles

==>java we use to import packages.

==>js we require the module(which module u require)

==>require() is used to import/include a module into js file

var ref=require('module-name')//require() gives reference of the attached module.

==>with the help of ref we can call any function of this module.

ex1-require('http').demo()

ex2-var ref=require('http');

ref.demo();

==>create our own server.

requirements are server should run on ip address, but we dont have any realtime server.

treat our system as our server.if our local system is acting like a server then we get ip address 127.0.0.1 (localhost is the domain name)

==>localhost is the domain name for reaching out to ip address 127.0.0.1

==>every ip address has some domain name.

==>nodejs will give ip address as 127.0.0.1, if we want to send request from the brower to the our server ,so on which ip address the request should be pass i.e 127.0.0.1/localhost

==>when u r sending request to your own server only writing ip address is not enough,we have defined port no.

3 web application is running on 127.0.0.1 so i want to send request to only 1 web application.

==>how to uniquly indentify webappliation by using port number.

==>we have to assign unique port number.

==>port number is just a unique identity given to a process running on a server so that we can know in prior how we can reach to specific application.so its developers responsilbilty to assign a port number.

127.0.0.1:port-no(proper way of sending request to our own server)

http://127.0.0.1:4500

==>dont use 8000,8080(tomcat server),3600(mysql),3000(react),27017==>mongodb

===========================================================================

CREATE A SERVER BY USING NODEJS

==>1 predefined module i.e http module .in this module,we will get predefined functions to create server

==>cmd prompt will terminate after executing 1 instruction,server will going to stop in between,server will continously listen to your request and give response.i.e use listen()

const http=require('http')

const server=http.createServer((req,res)=>{

//read the request,url use request obj

//send the response use response obj

});//createServer() will create a server ,return reference of newly created server. we can also start /stop the server

//this function will be executed as a callback whenever request will be send by client

server.listen(port) //listen() will start the server on the mentioned port no and server will now continuous listen to the request.

=====================

==>createServer() returns a reference to newly created server.it accepts one call back function, and this callback will have 2 arguments.

This callback function will always be executed when there is a request from a client.

==>two argument:first one will be for request obj and 2nd will be for response obj.

==>listen() can take one callback function as the 2nd argument. this callback function will be executed when server has started listening to the requests.

so all the initial logic like db connection,opening few resources can be done inside this callback.

========================================================================================================

node node1.js // start the server ctr+c

browser-->127.0.0.1::4500 //send the request to url and portno

const http=require('http');

const server=http.createServer((request,response)=>{

response.write("hello world");

response.end();//this function will terminate the processing of server

});

server.listen(4500);

//ctr+c will break server execution.

===============================================================

const http=require('http');

const PORT=4500;

const server=http.createServer((request,response)=>{

response.write("hello world");

response.end();//this function will terminate the processing of server

});

server.listen(PORT,()=>{

console.log("SERVER IS LISTENING ON ${PORT}");

});

//ctr+c will break server execution.

=====================================================================

const http=require('http');

const PORT=4500;

const server=http.createServer((request,response)=>{

var x=4;

var y=8;

response.write(`sum of ${x} and ${y} is ${x+y}`);

response.end();//this function will terminate the processing of server

});

server.listen(PORT,()=>{

console.log("SERVER IS LISTENING ON ${PORT}");

});

//ctr+c will break server execution.

====================================================================

2)custom/userdefined module==>A module created by developer.

const http=require('http');

const myModule=require('./node2.js');

const PORT=4500;

const server=http.createServer((request,response)=>{

var x=4;

var y=8;

var z=myModule.sum(1,2);

response.write(`sum of ${x} and ${y} is ${z}`);

response.end();//this function will terminate the processing of server

});

server.listen(PORT,()=>{

console.log("SERVER IS LISTENING ON ${PORT}");

});

---------

(module)//this module cannot be accessible outside

function sum(x,y){

return x+y;

}

==============================================================================================

exports.sum=()=>{

return x+y;

}

===================================================================

REPL stands for Read Evaluate Print Loop, and it is a programming language environment (basically a console window) that takes a single expression as user input and returns the result back to the console after execution. The REPL session provides a convenient way to quickly test simple JavaScript code.

js is used only for REPL(read,evaluate(exp),print,loop) and language features(variable,datatypes,operators)

==>node REPL terminal provides a terminal to run node.js commands

==>you can also run REPL

==>to open REPL terminal

c:\> node

var x=10

var y=20;

var z=x+y

console.log(z)

=========================================================

node.js programs

==>every node.js program is a javascript program stored in .js file

==>you can compile and run by using node compiler

ex:

node1.js

console.log("hello");

C:\FILE>node node1.js

hello

=================================================

var msg="hello node js";

console.log(msg);

C:\FILE>node node1.js