

SCS1620 User Interface Technologies

Unit IV – Node JS (Server Side JS Framework)

Introduction - What is Node JS – Architecture – Feature of Node JS - Installation and Setup - Creating web servers with HTTP (Request & Response) – Event Handling - GET & POST implementation - Connect to SQL Database using Node JS – Implementation of CRUD Operations.

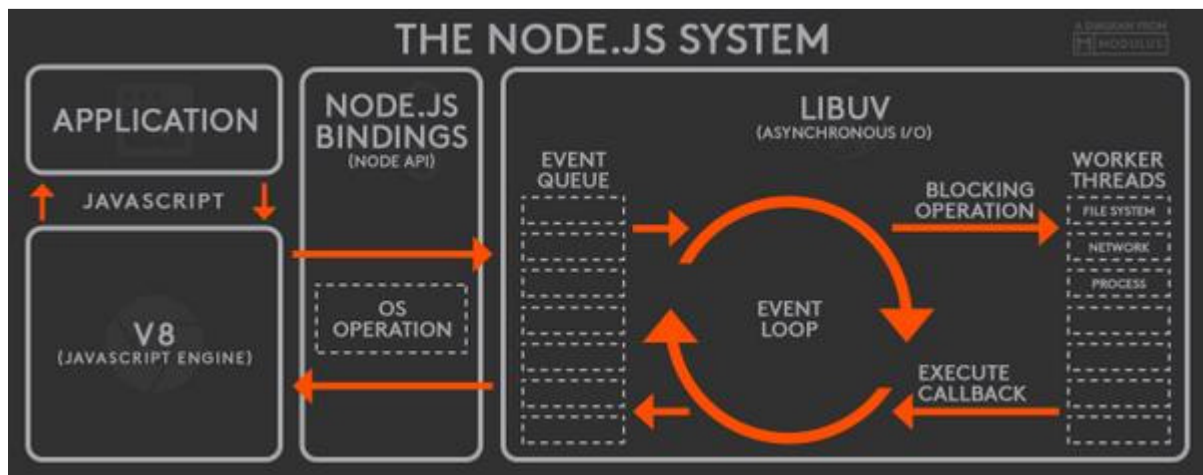
Introduction

- Node.js was developed by Ryan Dahl in 2009
- Node.js is an open source server framework
- Node.js allows you to run JavaScript on the server
- Node.js runs on various platforms
- Node.js uses asynchronous programming
- Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine)
- Node.js = Runtime Environment + JavaScript Library

Official definition of Node.js

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Architecture

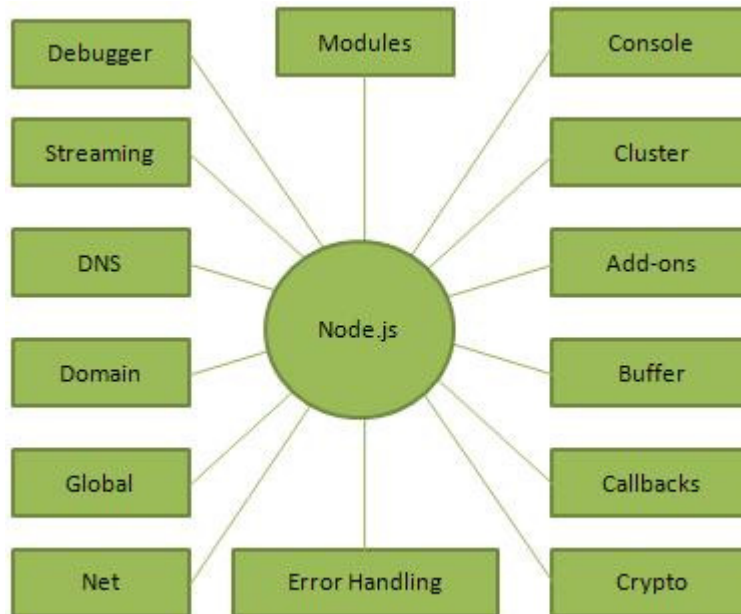


A common task for a Web server can be to open a file on the server and return the content to the client. For example php, asp or jsp handles a file request in the following sequence:

- Sends the task to the computer's file system.
- Waits while the file system opens and reads the file.
- Returns the content to the client.

- Ready to handle the next request.

Components of Node.js



Whereas, Node.js handles a file request:

- Sends the task to the computer's file system.
- Ready to handle the next request.
- When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the waiting, and simply continues with the next request.

Node.js runs single-threaded, non-blocking, asynchronously programming, which is very memory efficient.

Node.js can perform:

- Node.js can generate dynamic page content
- Node.js can create, open, read, write, delete, and close files on the server
- Node.js can collect form data
- Node.js can add, delete, modify data in your database

Download and install node.js from <https://nodejs.org>. (Version 9.8.0)

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.end('Welcome to CSE!');
}).listen(9999);
```

// To initiate the nodejs code execute in command line mode:

```
// node example1.js
```

```
// To execute in a browser:  
// http://localhost:9999/example1.js
```

Node JS Modules

- Modules are a set of functions to be consumed in an application.
- Modules can be user defined or built in (several modules are available).
e.g. `var tmp=require('http');`
“http” is a module for processing http request and responses.
To load a module, the function “require” is used.

Example

To create a user defined module (save as module1.js)

```
exports.myDateTime = function () {  
  return Date();  
};
```

To consume the module

```
var http = require('http');  
var dt = require('./module1'); // location of the module.
```

```
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.write("The date and time are currently: " + dt.myDateTime());  
  res.end();  
}).listen(9999);
```

http Module

- Node.js has a built-in module called HTTP, which allows Node.js to transfer data over the Hyper Text Transfer Protocol (HTTP).
- The HTTP module can create an HTTP server that listens to server ports and gives a response back to the client.
- The function passed into the **http.createServer()** method, will be executed when the client tries to access the computer on port 9999.
- If the response from the HTTP server is supposed to be displayed as HTML, the HTTP header should be specified.
- The function passed into the `http.createServer()` has a **req** argument that represents the request from the client, as an object (**http.IncomingMessage object**).
- This object has a property called "url" which holds the part of the url that comes after the domain name

Example 3:

```
var http = require('http');
```

```

var url = require('url');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write(req.url);
  res.write("<br>");
  var q = url.parse(req.url, true).query;
  var txt = q.year + " " + q.month;
  res.end(txt);
}).listen(9999);

```

// To execute: <http://localhost:9999/example3.js?year=2018&month=April>

NodeJS Package Manager (NPM)

- NPM is a package manager for Node.js packages, or the required modules
- www.npmjs.com hosts thousands of free packages to download and use.
- The NPM program is installed on your computer along with Node.js
- A package in Node.js contains all the files you need for a module.
- Modules are JavaScript libraries you can include in your project.
- To download and install a package:
 - `npm install module-name`
 - example: `npm install upper-case`

Example 4:

```

var http = require('http');
var uc = require('upper-case');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/html'});
  res.write(uc("cse!"));
  res.end();
}).listen(9999);

```

Events

Node.js is perfect for event-driven applications.

Objects in Node.js can fire events, like the `readStream` object fires events when opening and closing a file

Example:

```

var fs = require('fs');
var readStream = fs.createReadStream('./example4.js');
/*Write to the console when the file is opened:*/
readStream.on('open', function () {
  console.log('The file is open');
});

```

Node.js has a built-in module, called "Events", which performs create-, fire-, and listen for- user defined events.

Include the "events" module

Event properties and methods are an instance of an EventEmitter object.

To be able to access these properties and methods, create an EventEmitter object:

To trigger an event, the method emit() is triggered.

```
var events = require('events');
var EventEmitter = new events.EventEmitter();

//Assign the event handler to an event:
eventEmitter.on('name of the event', eventHandlerName);

//Fire the 'scream' event:
eventEmitter.emit('name of the event');
```

Database with Node.js

Install XAMPP Server. Server to run in localhost and use the port 3306

Module: mysql

Server: xampp

Example: Check Connectivity:

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root"
});
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
});
```

Example: Create Database:

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root"
});
con.connect(function(err) {
```

```
if (err) throw err;
console.log("Connected!");
con.query("CREATE DATABASE nodejsdb", function (err, result) {
  if (err) throw err;
  console.log("Database created");
});
});
```

Example 3: Create Table:

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root",
  database: "nodejsdb"
});
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "CREATE TABLE programmes (id INT AUTO_INCREMENT PRIMARY
KEY, school VARCHAR(25), deptt VARCHAR(50))";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Table created");
  });
});
```

Example 4: Insert Records (a):

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root",
  database: "nodejsdb"
});
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql1 = "INSERT INTO programmes (school, deptt) VALUES ('Computing',
'CSE')";
  con.query(sql1, function (err, result) {
    if (err) throw err;
    console.log("1 record inserted");
  });
});
```

```

var sql2 = "INSERT INTO programmes (school, deptt) VALUES ('Computing', 'IT')";
con.query(sql2, function (err, result) {
  if (err) throw err;
  console.log("1 record inserted");
});
});

```

Example: Insert Multiple Records (b):

```

var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root",
  database: "nodejsdb"
});
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "INSERT INTO programmes (school, deptt) VALUES ?";
  var values = [
    ['Electronics','ECE'],
    ['Electronics','ETCE'],
    ['Electronics','EIE'],
    ['Electronics','EEE']
  ];
  con.query(sql, [values], function (err, result) {
    if (err) throw err;
    console.log("Number of records inserted: " + result.affectedRows);
  });
});

```

Result Object:

```

fieldCount
affectedRows
insertId
serverStatus
warningCount
message -> Records, Duplicated, Warnings
protocol41
changedRows

```

Example: Select

```

var mysql = require('mysql');
var con = mysql.createConnection({

```

```

    host: "localhost",
    user: "root",
    password: "root",
    database: "nodejsdb"
  });
  con.connect(function(err) {
    if (err) throw err;
    console.log("Connected!");
    // SELECT * FROM programmes
    con.query("SELECT school, deptt FROM programmes", function (err, result, fields)
    {
      if (err) throw err;
      console.log(result)
      console.log(result[2].school);
      console.log(fields);
      console.log(fields[1].name);
    });
  });
});

```

Example: Select Query:

```

var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root",
  database: "nodejsdb"
});
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  con.query("SELECT school, deptt FROM programmes where deptt='CSE'", function
(err, result, fields) {
    if (err) throw err;
    console.log(result)
  });
  con.query("SELECT school, deptt FROM programmes WHERE deptt LIKE 'E%'
ORDER BY deptt", function (err, result, fields) {
    if (err) throw err;
    console.log(result)
  });
  var deptt="IT";
  var sql = "SELECT school, deptt FROM programmes WHERE deptt=" +
mysql.escape(deptt);
  con.query(sql, function (err, result) {

```



```

if (err) throw err;
console.log(result);
});
var deptt="IT";
var school="Mechanical"
var sql = "SELECT school, deptt FROM programmes WHERE school = ? OR deptt
= ? ORDER BY deptt DESC";
con.query(sql, [school, deptt], function (err, result) {
if (err) throw err;
console.log(result);
});
});

```

Example: Delete Query

```

var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root",
  database: "nodejsdb"
});
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  con.query("DELETE FROM programmes where deptt='AERO'", function (err, result)
  {
    if (err) throw err;
    console.log("Records Deleted: " + result.affectedRows);
  });
});

```

Example: Update Query

```

var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root",
  database: "nodejsdb"
});
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  con.query("UPDATE programmes SET deptt='Computer Science and Engineering'
where deptt='CSE'", function (err, result) {

```

```

    if (err) throw err;
    console.log("Records Updated: " + result.affectedRows);
  });
  con.query("UPDATE programmes SET school='Mechanical' where
school='Mechanical'", function (err, result) {
    if (err) throw err;
    console.log("Records Updated: " + result.affectedRows);
  }); });

```

Example: Drop Table

```

var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root",
  database: "nodejsdb" });
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "DROP TABLE programmes";
  //var sql = "DROP TABLE IF EXISTS programmes";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Table deleted");
  });
});
});

```

JOINS

```

var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "root",
  database: "nodejsdb" });
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql="SELECT programmes.school, student.regno, student.name FROM
programmes JOIN student ON student.deptt = programmes.deptt";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log(result);
  }); });

```