NodeJS - Learning Guide

* Prerequisites :
  + Basic understanding of JavaScript
  + Some understanding of other web technologies such as HTML, CSS, AJAX, etc.
* Installation Steps: (**Please skip this step if you are using TopGear MEAN VIDI as it is pre-installed with npm and mongodb**)
  + - Download the Windows installer from the ***Nodes.js***® web site.
    - Run the installer (the .msi file you downloaded in the previous step.)
    - Follow the prompts in the installer (Accept the license agreement, click the NEXT button a bunch of times and accept the default installation settings).
    - To ensure that you have node.js installed, open a command prompt and make sure the

following commands exist and are at least the versions below:

* + - > node –version
    - npm --version
* Overview:
* This training on ‘NodeJS’ enables the developers to work with NodeJS, and developing Server Side programs using NodeJS

Reference links:

|  |  |  |
| --- | --- | --- |
| Topic | Sub Topics | Reference Links |
| Overview of Java Script | * Basics of Java Script * Data Types * Object and Array creation * Functions & Inner Functions * Using V8 Engine of Chrome | * <http://www.tutorialspoint.com/javascript/javascript_overview.htm> * <http://www.w3schools.com/js/js_functions.asp> * <https://davidwalsh.name/javascript-functions> |
| Introduction to Node JS | * Features of Node JS * Node REPL * NPM – node package manager | * <http://www.tutorialspoint.com/nodejs/nodejs_introduction.htm> * <http://www.tutorialspoint.com/nodejs/nodejs_npm.htm> |
| Node Fundamentals | * Modules * Call backs * Events * Error Handling | <http://www.tutorialspoint.com/nodejs/nodejs_callbacks_concept.htm> |
| Global Objects | * console * Process * setTimeout() * clearTimeout() * setInterval() | * <https://nodejs.org/api/globals.html> * <https://nodejs.org/api/process.html> |
| Utility Modules | * OS * fs * Path * Net * Domain… | <https://nodejs.org/api/fs.html>  <http://www.sitepoint.com/accessing-the-file-system-in-node-js/> |
| Web module | * Using ‘http’ module   + Creating Server   + Creating a client | https://nodejs.org/en/docs/guides/anatomy-of-an-http-transaction/ |
| Working with Data Base | * Working with ‘MySQL’ RDBMS * Working with ‘MongoDB’ -NoSQL data base. | * <http://www.sitepoint.com/using-node-mysql-javascript-client/> * <https://codeforgeek.com/2015/01/nodejs-mysql-tutorial/> * <https://docs.mongodb.org/getting-started/node/client/> |

**Expected deliverables:**

1. Solutions for hands on assignments.
2. NodeJS based Sample Project.

(**Users can check in these deliverables using two separate folders “Assignment solutions” and “NodeJS project” in their GITLab repository**)

1. Hands on assignments :
   1. Create an Object type in Java Script and add functions & variables to it
   2. Create an array, add variables and function to it through push()
   3. Create a ‘js’ file with code that provides implementation for ‘pwd’ command from ‘Node’ shell.
   4. Creata NodeJS based script file, that reads the name of the directory from the command line arguments and displays the list of directory contents (using fs module)
   5. Create a Node JS script that reads the file name from console and displays the contents of the file
      1. Synchronous mode
      2. Asynchronous mode
   6. Create a Node JS Script that displays a message through loop, with delay in between the iterations
      1. Using setTimeOut()
   7. Create a Node JS Script, using ‘http’ module that downloads the content from a web page to a file
      1. E.g. to download the google home page
   8. Create a simple HTTP server that responds to requests with a simple HTML response. (using http module)
   9. Create a program to accept client requests on a socket (using net module) and further to create client socket for communication
      1. Create Server socket that listens at a port number and creates sockets for every client request.
      2. Send a message from any of the client (socket) and through code the message should be propagated to all the sockets which are active
      3. Test it through ‘telnet’, to see the message echo.
   10. Create a Node JS script that uses ‘mongodb’ module to read the data from a table in ‘mongodb’ database and display the details on console.
   11. Create a Node JS Script that uses ‘mongodb’ specific modules and provides ‘CRUD’ operations on a Collection of MongoDB database.
2. Sample Project
   1. Create a simple ‘HTML’ form to accept the details of a new Student
   2. Submit the details to a ‘Node JS’ Http Server application
      1. Use available ‘Data Base’ (mongodb) to create a table for storing ‘students’ details.
      2. Use NodeJS for taking the data from Http Request and send the data to DB for insertion into the table.
   3. Send the appropriate response back to the client (browser) indicating the status of operation.

Response with status

DB

Student Registry (HTML)

(HTML)

Node JS

Server App