WORKSHOP

ON

NODE JS

[BUILDING REST API’S USING NODE]

Revision History

Ver. #	Description of change	Sections Affected	Created/ Modified By	Reviewed By	Approved By	Date of issue
0.1	Initial Draft	All	Meganathan	Sridhar Vasudevan		18/12/2018

Table of Contents

1. Introduction

Node.js is a very powerful JavaScript-based framework/platform built on Google Chrome's JavaScript V8 Engine. Node.js is open source, completely free. Node.js is a server side javascript platform, asynchronous and cross-platform runtime environment. Node.js is highly scalable, super fast backend development framework. Node has largest ecosystem of open source library available. Node is a runtime environment for executing javascript code.

1.1 Node History

Before Node, Javascript runs only in browser. Every browser have a javascript engine to run javascript code. Different browser use different script engine: Chrome use V8, IE use Chakra, Firefox use Spider Monkey. Because of variety of JS Engine, javascript code may give different results on different browser. Till 2009 JS only able to run in browser.

Ryan Dhal creator of Node come with a brilliant idea, he through it will be great if we can run javascript outside the browser. He took Google V8 engine which is the fastest JS engine and embedded that in a C++ program that is Node.

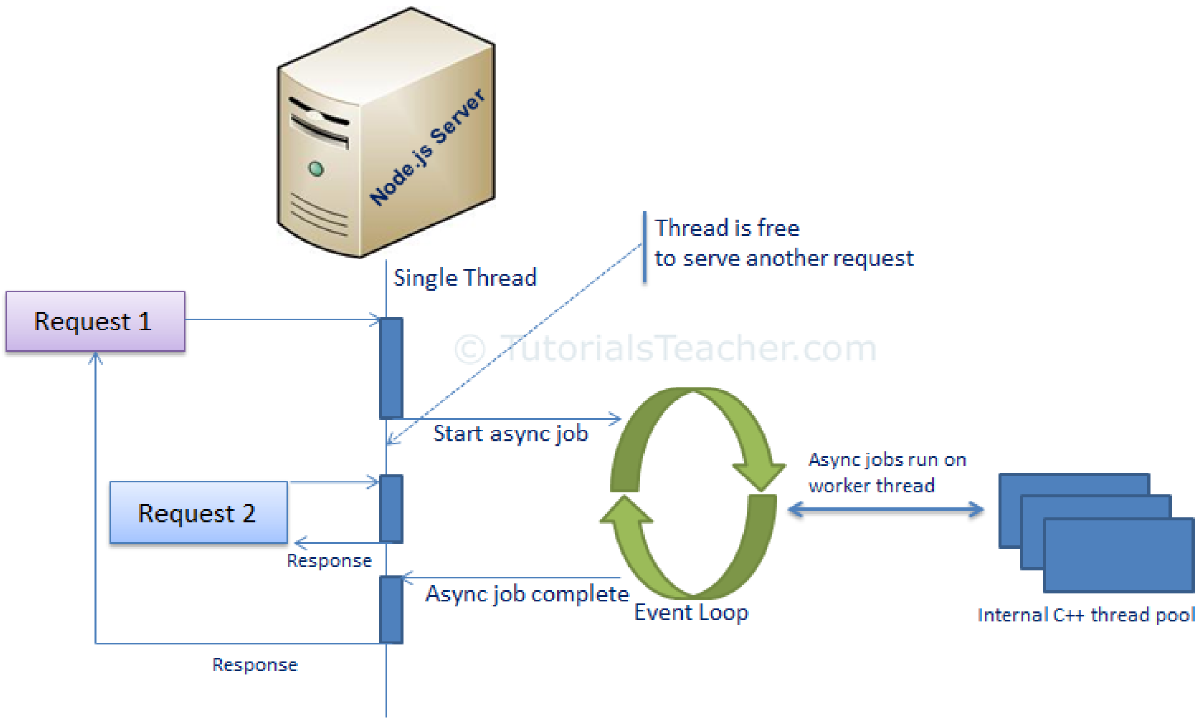
Node is not a programming language, its only a runtime environment for executing JS code.

1.2 Prerequisites

You should have a basic understanding of Node with Express framework as we are going to develop REST API’s.

2. Node JS Architecture

Node.js runs in a single process and the application code runs in a single thread and thereby needs less resources than other platforms. All the user requests to your web application will be handled by a single thread and all the I/O work or long running job is performed asynchronously for a particular request.



Node applications are asynchronous by default, we don't have to do anything extra. In Node, single thread is handling all requests. When a request came single thread will use, if there any DB / file system operation thread will not wait response, if there any new request same thread will fulfill another request and so on. Node is continuously monitoring 'Event Queue' in background for for handling message back to main thread.

Node may not use CPU intensive applications like Video encoding, image processing. Node only use for data intensive and network applications.

3. Node Modules

At the core of Node, we have the concept called Module. Every files in the node is considered as module, the variables and function defined are scope to that module, not available outside to the module. You can export an object or function using export module. You can import a module using ‘require’ keyword.

Node provides different built-in modules. Explore the below link to understand the list of built-in modules:
https://nodejs.org/dist/latest-v8.x/docs/api/

Understand the modules such as Console, File System, OS, Events, HTTP, Crypto in details which will help to complete the workshop.

4. HTTP Module

This module helps to create a web-server that listens for http requests on a given port. We can create backend system using this module for Angular, React or mobile applications.

The HTTP module in Node.js are designed to support many features of the protocol which have been traditionally difficult to use. In particular, large possibly chunk-encoded, messages.

Create a sample project using http module with all http methods and share the code.

Use the below link for more information:

https://nodejs.org/dist/latest-v8.x/docs/api/http.html

http://www.tutorialsteacher.com/nodejs/create-nodejs-web-server

https://www.youtube.com/watch?v=TlB_eWDSMt4&t=875s

5. Express Framework

Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications. Following are some of the core features of Express framework –

- Allows to set up middlewares to respond to HTTP Requests.
- Defines a routing table which is used to perform different actions based on HTTP method and URL.
- Allows to dynamically render HTML Pages based on passing arguments to templates.

Use the below link for more details about express.js features

https://www.npmjs.com/package/express
https://expressjs.com/en/4x/api.html
https://www.tutorialspoint.com/nodejs/nodejs_restful_api.htm
https://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm

Let’s create a new web project in Node using express.

6. Automatic Changes Management

Automatic monitoring and updating change to running server using nodemon package.

Use the below link for more information on how to use nodemon:

https://www.npmjs.com/package/nodemon

7. Authenticate API using JWT

JSON Web Tokens are an open, industry stardard RFC 7519 method for representing claims securely between two parties.

JWT.IO allows you to decode, verify and generate JWT.

JWT authentication is becoming very popular these days. The traditional authentication uses cookies and sessions. With the advent of Single Page Applications (SPA) and Microservices, there is a need to look beyond the sessions. Any token based authentication serves that purpose. JWT is a type of token-based authentication. For every single request from a client to the server, a token is passed for authentication. It supports the stateless API calls.

Use the below links for more information:

https://scotch.io/tutorials/authenticate-a-node-js-api-with-json-web-tokens

https://medium.com/dev-bits/a-guide-for-adding-jwt-token-based-authentication-to-your-single-page-nodejs-applications-c403f7cf04f4

8. Request Validation

Object validation on server side is a necessity for public web services. The realisation of this necessity became Joi - the most widely adopted package for object schema descriptions and validation. Joi allows us to create blueprints of Javascript objects that ensure that we process and ultimately accept accurate data.

Use the below linke for more information:

https://www.npmjs.com/package/joi

https://medium.com/@rossbulat/joi-for-node-exploring-javascript-object-schema-validation-50dd4b8e1b0f

https://scotch.io/tutorials/node-api-schema-validation-with-joi

9. Node and MySQL

Node.js and MySQL is one of the necessary binding needed for any web application. MySQL is one of the most popular open source database in world and efficient as well. Almost every popular programming language like Java or PHP provides driver to access and perform operations with MySQL.

Use the below link for more information:
https://www.npmjs.com/package/mysql#introduction

10. Reference Links

10.1 Basic Understanding of Node JS please watch below link

https://www.youtube.com/watch?v=TlB_eWDSMt4&t=1286s

10.2 Express library for Rest API

https://www.youtube.com/watch?v=pKd0Rpw7O48

11. Assessment Model

11.1 Assessment Evaluation Criterias:

Assessment should be measured based on sample projects, POC and open discussion between participants.

As per the dates mentioned on the plan, share the sample projects developed to me.

At the end of workshop, each participants should develop a POC based on what you learned during the workshop.

The POC details will be shared to each individual and each one of you should share the code as per the date mentioned on the plan.

11.2 Workshop Duration: 64 hours

The duration includes learning the concepts, discussions, creating sample projects and final POC.