

ASSIGNMENT 2

Node JS



Nagarro Campus Learning Program

Objective

Purpose of the document is to go through below items:

- Overview of NodeJS
- Overview of MongoDB
- Creating a simple server application for the problem statement mentioned.

Assignment requirements

- Create a server application, that can be used for registering students and companies for a placement event in an institute.
- The solution should use NodeJS and MongoDB.
- The server application should expose following APIs
 - Add, Remove, and Edit for 'Students' in the college.
 - Register and Unregister of 'Company' for the placement event in the college.
 - Register and Unregister a 'Student' for a 'Company' in the college.
- Above APIs should be completed in all respect including database.
- Additionally –
 - Implementing API Data validation,
 - Find a suitable data validation library for NodeJS.
 - Explore the functionality and how to use.
 - Implement the validation functions in the assignment for validating and sanitizing the data that application APIs handle.
 - Implementing Error handling in the application
 - Find a suitable error handling library for NodeJS
 - Explore the functionality and how to use.
 - Implement the error handling, with proper **Response codes** and intuitive **Error messages**.
 - Implementing Logging in the application
 - Find a suitable logging library for NodeJS.
 - Explore the functionality and how to use.
 - Implement logging, with proper timestamp, log type (Error, Warning or Information) and an error message.

How to do?

- This document will walk through basic steps, describing how to solve the problem.
- Also the document will,
 - Refer to some links on internet for the understanding required to complete this assignment.
 - Refer to advanced / exhaustive material and practice questions for interested students. Please note, going through and implementing such advanced topics (other than the problem statement) is not mandatory to complete the assignment.

Submission guidelines

- Student should be submitting all files that constitute the solution of the problem statement mentioned in the document.
- In addition if there are any enhancements that the student has tried, should also be part of the submission.

- Ensure the files are named appropriately.
- Ensure folder structure used is appropriate and as simple as possible.
- All the code is properly commented to make them readable.

Assignment

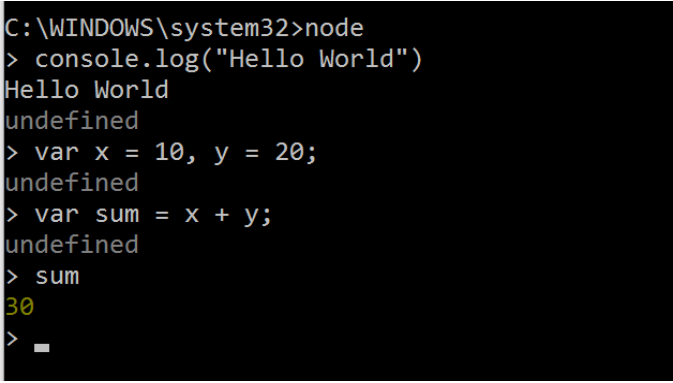
Step 1: Understanding NodeJs

- Node.js is an open-source, cross-platform JavaScript runtime environment for developing a diverse variety of tools and applications.
- Node.js has an event-driven architecture capable of asynchronous I/O.
- TODO: Explore and understand the terms:
 - Open Source, Cross Platform, Event driven systems, Synchronous and Asynchronous methods.
- Links,
 - <https://www.techopedia.com/definition/27927/nodejs>
 - <http://whatistechtarget.com/definition/Nodejs>

Step 2: Installing NodeJS

- Get and Install NodeJS from, <https://nodejs.org/en/download/>
- Pick your IDE for development from, <http://codecondo.com/best-ide-for-node-js/>
Note: This step is optional, students can use Notepad/Notepad++ to complete this assignment.
- Example video tutorial for installation, <https://www.youtube.com/watch?v=tlntE8fe6u4>

Step 3: Testing NodeJS installation

- Run few commands,

```
C:\WINDOWS\system32>node
> console.log("Hello World")
Hello World
undefined
> var x = 10, y = 20;
undefined
> var sum = x + y;
undefined
> sum
30
> _
```
- Explore more,
 - <http://theholmesoffice.com/testing-your-node-js-install/#>
 - <http://blog.teamtreehouse.com/install-node-js-npm-windows#>

Step 4: Understanding MongoDB

- MongoDB is a free and open-source cross-platform document-oriented database program.
- Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas.
- TODO: Explore and understand the terms:
 - NoSQL databases, document-oriented data model, JSON.
- Links,
 - <http://searchdatamanagement.techtarget.com/definition/MongoDB>
 - <https://www.mongodb.com/what-is-mongodb>

Step 5: Installing MongoDB

- Assignment can be done setting up mongoDB in either of two ways mentioned below,
 - Setup Mongo Database Online
 - Setup Mongo Database Offline
- Both the ways will be explained briefly in next few pages. This would help understand the setup and choose one for the assignment

Step 6: Understanding Setup Mongo Database Offline

- Install MongoDB from,
 - <https://www.mongodb.com/download-center#community>
- Install, Start and test MongoDB,
 - <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/>
- Create Database,
 - https://www.tutorialspoint.com/mongodb/mongodb_create_database.htm
 - https://www.tutorialspoint.com/mongodb/mongodb_create_collection.htm
- Example video tutorial for installation
 - https://www.youtube.com/watch?v=1yZj_9VWMdE

Step 7: Understanding Setup Mongo Database Online

- There are MongoDB hostings available online which can be used to maintain the database and connect from anywhere.
- Below are few examples, get registered and host your database.
 - MongoLab
 - <https://mlab.com/>
 - MongoHQ
 - <https://www.compose.com/>
 - ObjectRocket
 - <http://objectrocket.com/>
- Links
 - <https://www.quora.com/What-is-the-best-mongodb-hosting>
 - <http://mrdanadams.com/2012/mongohq-mongolab-mongodb-customer-service/#.WFd5NvI942w>
- Video tutorial as an example,
 - <https://www.youtube.com/watch?v=4-xXnj3Jxz8>

Step 8: Setup npm

- Node Package Manager (NPM) provides two main functionalities –
 - Online repositories for node.js packages/modules which are searchable on search.nodejs.org
 - Command line utility to install Node.js packages, do version management and dependency management of Node.js packages.
- Links
 - <http://stackoverflow.com/questions/31930370/what-is-npm-and-why-do-i-need-it>
 - https://www.tutorialspoint.com/nodejs/nodejs_npm.htm
- Video tutorial on how to setup npm for your assignment,
 - <https://www.youtube.com/watch?v=LdxaLDMKM3Q>

Step 9: Install required packages for your assignment using npm

- Express,
 - Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.
 - Explore more,
 - <http://expressjs.com/>
- Mongoose,
 - Mongoose provides a straight-forward, schema-based solution to model your application data. It includes built-in type casting, validation, query building, business logic hooks and more, out of the box.
 - Explore more,
 - <http://mongoosejs.com/>
- In a nutshell, your code would consist of below blocks,
 - Models
 - Below are the business models we will be working with in the assignment,
 - Student
 - Company
 - Server
 - Represents the server that should be hosted, so that outside world can use the application

Step 10: Construct Student Model

- Student
 - Create a file “student.js”
 - Responsibility of Student are as follows,
 - Define the schema.


```
var studentSchema = new mongoose.Schema({
  name: String,
  department: String,
  rollno: Number,
  cgpa: Number
});
```
 - Export the model


```
module.exports = restful.model('tblstudent', studentSchema);
```
- Company
 - Create Company model in the same way as explained above.
- Todo:
 - Explore the functions mentioned above. Study more on new terms you encounter and mention them all in the assignment.

Step 11: Construct Router

- Create a file “api.js”
- Responsibility of the router is as follows,
 - Create a router object,

```
//dependencies
var express = require('express');
var router = express.Router();
```

- To define the application routes. In order to define the routes it has to import the models we have created. Note the APIs being registered for the model ('get', 'put', 'post', 'delete'),

```
//register student
var student = require('./student');
student.methods(['get', 'put', 'post', 'delete']);
student.register(router, '/student');
```

- Export the router

```
//return router
module.export = router;
```

- TODO: Explore the functions mentioned above. Study more on new terms you encounter and mention them all in the assignment.

Step 12: Construct Server

- Create a file “server.js”
- Below are the responsibilities of the server,
 - Configure the server,

```
var app = express();
var bodyParser = require('body-parser');
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
app.listen(1000);
```

- Import the router object, and use in the application,

```
app.use('/api', require('./api'));
```

- Configure mongoose,

```
var mongoose = require('mongoose');
mongoose.connect('My-Db-Name');
```

- TODO: Explore the functions mentioned above. Study more on new terms you encounter and mention them all in the assignment.

Step 14: Testing

- Implement following APIs –
 - Add, Remove, and Edit for ‘Students’ in the college.
 - Register and Unregister of ‘Company’ for the placement event in the college.
 - Register and Unregister a ‘Student’ for a ‘Company’ in the college

Step 15: Testing

- From the hosted location (such as <http://localhost/api/student>), test the created Apis using rest clients such as postman,
- Links,
 - <https://www.youtube.com/watch?v=O6la-NJYiu8>

Step 16: Additional important implementations

- Implementing API Data validation,
 - Find a suitable data validation library for NodeJS.
 - Explore the functionality and how to use.
 - Implement the validation functions in the assignment for validating and sanitizing the data that application APIs handle.
- Implementing Error handling in the application
 - Find a suitable error handling library for NodeJS
 - Explore the functionality and how to use.
 - Implement the error handling, with proper **Response codes** and intuitive **Error messages**.
- Implementing Logging in the application
 - Find a suitable logging library for NodeJS.
 - Explore the functionality and how to use.
 - Implement logging, with proper timestamp, log type (Error, Warning or Information) and an error message.

*For help in finding the suitable libraries, you can explore link mentioned in **More References** section.*

More References

- Below are the more references where you can explore nodeJS
 - <http://stackoverflow.com/questions/2353818/how-do-i-get-started-with-node-js>
 - <https://www.airpair.com/javascript/node-js-tutorial>
 - <https://www.codementor.io/mattgoldspink/tutorials/nodejs-best-practices-du1086jja>
 - <https://scotch.io/tutorials/creating-a-single-page-todo-app-with-node-and-angular>
 - <http://stackoverflow.com/questions/4088723/validation-library-for-node-js>
 - <https://nodejs.org/api/errors.html>
 - <https://www.loggly.com/blog/node-js-error-handling>
 - <https://expressjs.com/en/guide/error-handling.html>
 - <https://blog.risingstack.com/node-js-logging-tutorial/>