Web Development: Module 2, Lesson 4  
REST Endpoints with Mocked Data Hands-On Lab

## Overview

In this lab, we'll implement in-memory (mocked data) RESTful API backend for a blog.

## Objectives

In this hands-on lab you will learn how to:

* Create file index.js and implement Node RESTful API server with CRUD using Express
* Use mocked data from the in-memory store, i.e., array hard-coded in the ([code/lesson4/lab/posts.json](https://github.com/MSFTImagine/computerscience/tree/master/Complimentary%20Course%20Content/Module2/code/lesson4/lab2))

## Prerequisites

The following are required to complete this hands-on lab:

* A text editor
* Windows PowerShell, Mac Terminal, or some other shell with node.js and npm installed
* You should have completed [Module 2 Lessons 1, 2, 3, and 4](https://github.com/MSFTImagine/computerscience/tree/master/Instructor-Led/Module2/Lessons) as well as the [Module 2 Lesson 2 Lab](https://github.com/MSFTImagine/computerscience/tree/master/Instructor-Led/Module2/Labs).

## Exercises

This hands-on lab includes the following exercises:

* Exercise 1: Node RESTful API server

## Exercise 1: Node RESTful API server

In this exercise, you will write a Node RESTful API server with CRUD using Express. Be sure to refer to [Module 2 Lesson 4](https://github.com/MSFTImagine/computerscience/tree/master/Complimentary%20Course%20Content/Module2/Lessons) throughout this exercise.

NOTE: The following resources should prove helpful.

* <http://expressjs.com/en/api.html>
* <http://webapplog.com/express-js-4-node-js-and-mongodb-rest-api-tutorial/>
* <http://chaijs.com/api/bdd/>

1. Create index.js file and implement GET, POST, PUT and DELETE for posts resource
2. You can use data from [code/lesson4/lab2/posts.json](https://github.com/MSFTImagine/computerscience/tree/master/Complimentary%20Course%20Content/Module2/code/lesson4/lab2)
3. You can use [code/lesson4/lab2/package.json](https://github.com/MSFTImagine/computerscience/tree/master/Complimentary%20Course%20Content/Module2/code/lesson4/lab2) to install dependencies with npm i
4. You can use curl or Postman to test or (recommended) use npm test on the command line to test with the test file [code/lesson4/lab2/express.test.js](https://github.com/MSFTImagine/computerscience/tree/master/Complimentary%20Course%20Content/Module2/code/lesson4/lab2) provided.
5. When all tests are passing — you have CRUD, compare your solution with express.js.

## Summary

In this hands-on lab, you learned how to:

* Implement basic RESTful API server
* Use Express