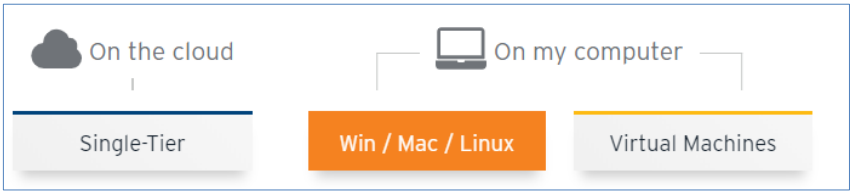
This document provides instructions on how to run the MEAN stack on your local machine to complete Module 4 Examples as an alternative to running on Google Cloud Platform. Ensure that you have Visual Studio Code installed on your local machine.

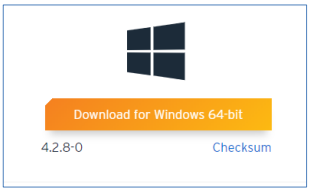
A. DOWNLOAD AND INSTALL MEAN SOFTWARE STACK ON YOUR COMPUTER

1 Go to the following URL: https://bitnami.com/stack/mean/installer

Under the "**On my computer section**" click on the "**Win / Mac / Linux**" button to download the MEAN stack.

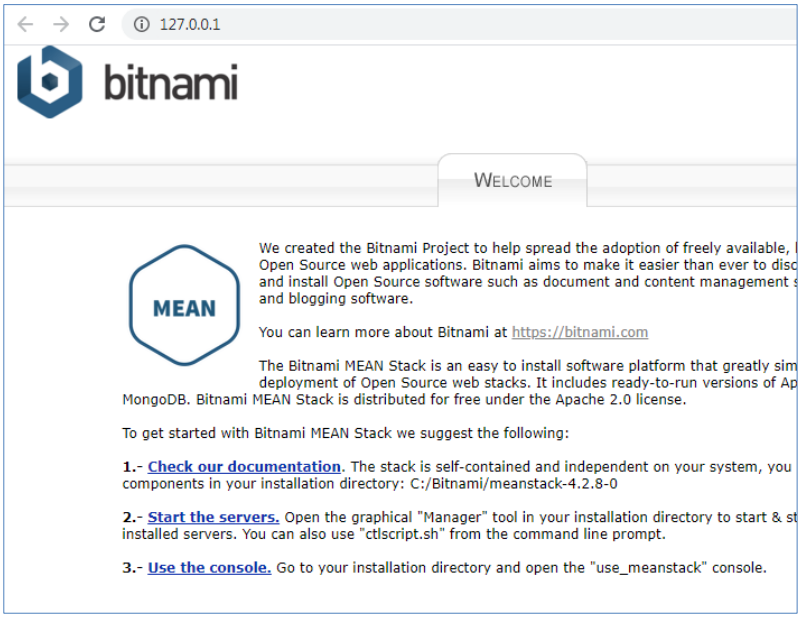


2 Choose the appropriate download version (e.g. Mac OS, Linux or Windows) and download the file and run on your machine. For this example, I will use Windows 64-bit.



3 Run the downloaded file and install the Bitnami MEAN stack***. This process will take few minutes.***

Once the installation completes, a browser instance will open and default to the URL http://127.0.0.1 . You should see the following page.



4 The Bitnami MEAN stack installs by default into the root of your default rive (e.g. C:). Launch the command prompt (Windows: Start | Accessories | Command Prompt or type cmd and click cmd.exe) or Terminal (Mac OS: Launchpad in the Dock | type Terminal and click Terminal). You should now see the command prompt. Change the directory to the default directory for Bitnami. In this example, it is

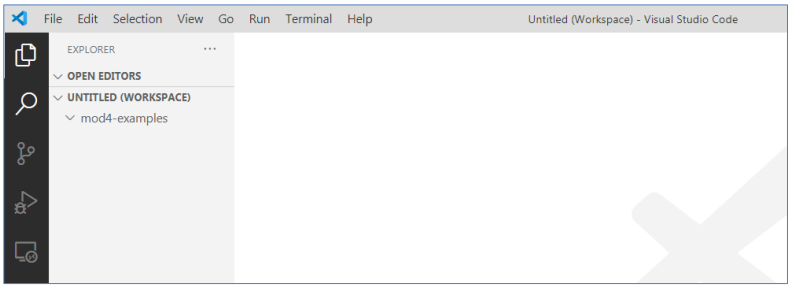
```
C:\Bitnami\meanstack-4.2.8-0>
```

Navigate to the folder /apache2/htdocs/ and create a new folder called **mod4-examples**

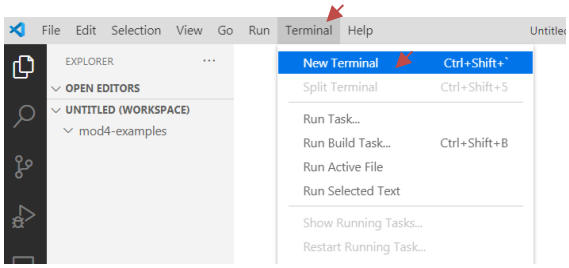
If you are running on Linux or Mac OS, please ensure that you execute the following command from the **htdocs** folder path:

chmod 757 mod4-examples

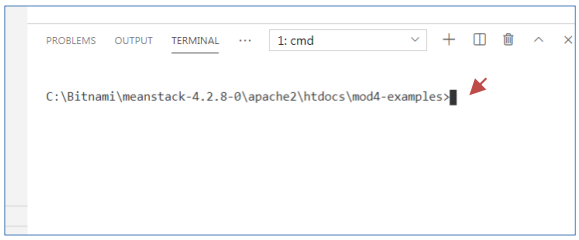
5 Launch **Visual Studio Code** and add the **mod4-examples** to the **workspace**:



6 Launch **Visual Studio Code** and add the **mod4-examples** to the **workspace**:



A window on the bottom part of the screen should now appear allowing you to run commands through this terminal window.



10 Enter the following command: **npm init**

This will allow you to initialize a Node.js application on this folder. Enter the following information:

- package name: **rand-gen-api**
- version (1.0.0): **1.0.0**
- description: **this is my first node.js application**
- entry point: (index.js): **app.js**
- test command: **build**
- git repository: **(press enter key to skip)**
- keywords: **node js random number generator**
- author: **enter your name**
- license: (ISC) **(press enter key, keep default)**

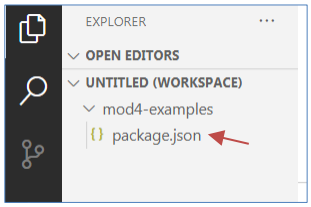
It will then prompt to check the entries you have made and asks if this OK? Type **yes** and press the enter key.

```
license: (ISC)
About to write to C:\Bitnami\meanstack-4.2.8-0\apache2\htdocs\mod4-examples\package.json:

{
  "name": "ran-gen-api",
  "version": "1.0.0",
  "description": "this is my first node.js application",
  "main": "app.js",
  "scripts": {
    "test": "build"
  },
  "keywords": [
    "node",
    "js",
    "random",
    "number",
    "generator"
  ],
  "author": "eyhab al-masri",
  "license": "ISC"
}


Is this OK? (yes) yes
```

npm should now create a **package.json** file on your workspace.

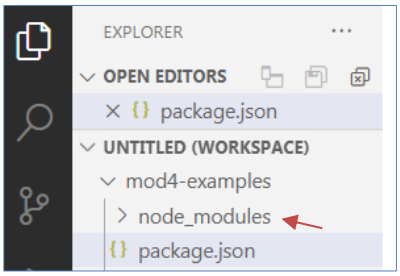


Open the **package.json** in the editor and change the "**scripts**" section as shown below:

11 From the terminal window, run the following command:

```
npm install express --save
```

This will setup the correct **express** version on your Node.js application. On the workspace, you will see a new folder called node-modules which contains all the dependencies required for running this application including **express**.



B. CREATING APP.JS

1 Create a new file called app.js in mod4-examples folder. Enter the following code:

```
// random number generator REST Service
// your name

// import required module
var express = require("express");
var app = express();
```

2 Then, define a **route** using a callback function that will be invoked when the user makes a HTTP request to the root folder. **Add** the following code:

```
// define a route using a callback function that will be invoked
// when the user makes a HTTP request to the root of the folder (URL)
// display some information about the REST Service
app.get('/', function (req, res) {
    res.status(200);
    res.send("<h1>This REST service will randomly generate numbers.</h1>");
    console.log("a request has been processed in / (root) ");
});
```

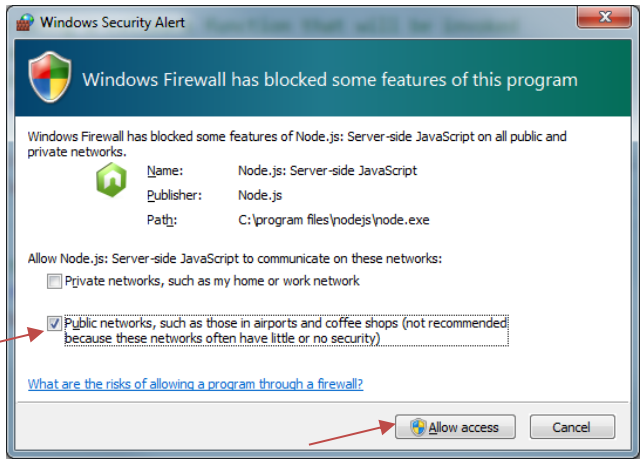
3 Then, we would like to create a listener on a port that will listen to incoming requests. **Add** the following code:

```
// enable a port to listen to incoming HTTP requests
app.listen(3000, function () {
    console.log("API version 1.0.0 is running on port 3000");
});
```

4 From the terminal window, run the following command to execute the Node.js application (ensure that you are in the mod4-examples folder):

```
C:\Bitnami\meanstack-4.2.8-0\apache2\htdocs\mod4-examples>node app.js
```

Your OS may prompt you to allow Node.js to access the port. You can allow access (example from Microsoft Windows OS below). Click **Allow access**.



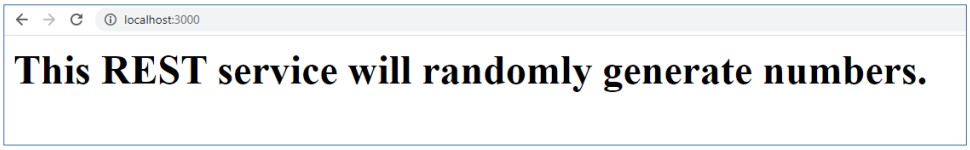
Your terminal now should be running the Node.js application and listening to incoming requests.

```
C:\Bitnami\meanstack-4.2.8-0\apache2\htdocs\mod4-examples>node app.js
API version 1.0.0 is running on port 3000
```

Open your browser (recommended Firefox) and enter any of the following URLs:

http://127.0.0.1:3000 or http://localhost:3000

You should now see the response from the route that we created.



To complete the rest of the example, please go to mod4-examples video and start at **minute 32**. In the video, there are some references to Google Cloud Platform which you can ignore since you are running the application locally.