

ITCS212 - Web Programming

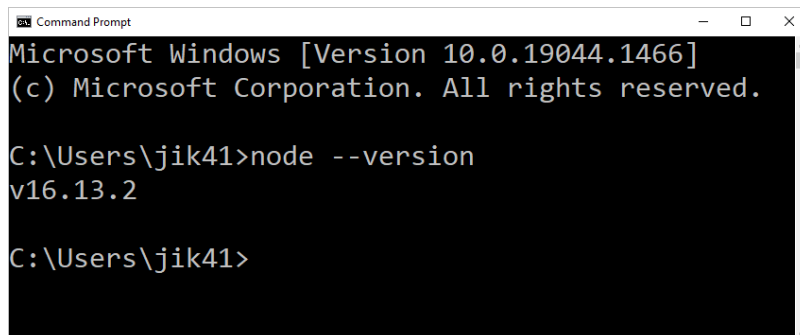
Lab 6: Node.js Part I

MUST READ:

- There are 2 questions to be completed: Q1 and Q2
 - Q2 is required to be submitted on MyCourses following the naming convention **by today's class time (Bangkok time)**. You must notify LAs **via the SOS** sheet to grade your work.
 - The SOS sheet will be opened by the instructor or LA (approximately the beginning of the lab). The student who wants to submit or ask questions shall put their name on the list. Do not put your name for reservation ahead of time.
 - The SOS sheet are here: [\[sec1\]](#)[\[sec2\]](#)[\[sec3\]](#)
 - **No late submission allowed**
-

Getting Node.js Installed

- Download and install Node.js version **16.13.2 LTS** (Go to <https://nodejs.org/en/>)
- To test if your computer has Node.js installed properly, open the command line interface type "`node --version`". It should show the version of the Node.js as follows



```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jik41>node --version
v16.13.2

C:\Users\jik41>
```

Q1 (No submission): Hello World in Node.js:

Your task is to implement a simple “Hello World” to users in Node.js. This question aims to help you set up and run your first Node.js file. More details can be found at https://www.w3schools.com/nodejs/nodejs_get_started.asp and https://www.tutorialspoint.com/nodejs/nodejs_first_application.htm)

1. Create the directory named with your section and ID, i.e., `C:\Users\[Your Computer Name]\[Lab06-SecY-XX88XXX]`
2. Open VSCode and set the current directory to the folder you created in Step 1
3. Create a new file, named "`L06-SecY-XX88XXX-q1.js`". Your directory should be like this:

```
Lab06-SecY-XX88XXX    (root)
|_L06-SecY-XX88XXX-q1.js  (current JS file)
```

4. In "`L06-SecY-XX88XXX-q1.js`", type the following code:

```
/* This is the first Hello World */  
console.log("Hello World!!");  
console.log("In the command line");
```

5. Run "L06-SecY-XX88XXX-q1.js", in the command line interface
 - a. Open the command line interface
 - b. Change the current directory to the one you just created
 - c. Type "node L06-SecY-XX88XXX-q1.js". Your output should be similar to the following screen

```
C:\[PATH]\Lab06-SecY-XX88XXX> node L06-SecY-XX88XXX-q1.js  
Hello World!!  
In the command line
```

No need to submit this question

Q2: Express Framework: Your task is to implement a simple Web server using Node.js Express Framework to show the different HTML pages to users (i.e. routing the page based on the request).

To start with Express, follow these steps to initialize your project:

1. Create a "L06-SecY-XX88XXX-q2" directory (Note: xxx is your ID and Y is your section), open the command line interface, and set the current work directory to this folder:

```
C:\[PATH]\L06-SecY-XX88XXX-q2>
```

2. use **npm init** to initialize your project. Note: **set L06-SecY-XX88XXX-q2.js as the entry point** (Note: xxx is your ID and Y is your section) and also put your name, your section and student ID as the author. After initialized, the structure of your directory should be similar to

```
L06-SecY-XX88XXX-q2  (root)  
|_ package.json
```

In package.json, the entry point and the author should be

```
{  
  "name": "l06-secY-xx88xxx-q2",  
  "version": "1.0.0",  
  "description": "",  
  "main": "L06-SecY-XX88XXX-q2.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit
```

```
1"
  },
  "author": "[Your Name], Section Y, 6X88XXX",
  "license": "ISC"
}
```

3. Install **express** using **npm install express --save** and **nodemon** using **npm install nodemon --save**. After installed, your the structure of your directory should be similar to

```
L06-SecY-XX88XXX-q2
> node_modules
|_ package-lock.json
|_ package.json
```

In **package.json**, the dependencies should be

```
{
  "name": "l06-secY-xx88xxx-q2",
  "version": "1.0.0",
  "description": "",
  "main": "L06-SecY-XX88XXX-q2.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit
1"
  },
  "author": "[Your Name], Section Y, 6X88XXX",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.2",
    "nodemon": "^2.0.15"
  }
}
```

Note ** Installing **nodemon** is optional. This is for your convenience only

4. Set up **nodemon** by putting the following code in the script part in the **package.json**: If you do not install **nodemon** in the previous step, set the value of "start" to "**node L06-SecY-XX88XXX-q2.js**". Note: you can remove the "test" script.

```
{
  "name": "l06-secY-xx88xxx-q2",
  "version": "1.0.0",
  "description": "",
  "main": "L06-SecY-XX88XXX-q2.js",
  "scripts": {
    "start": "nodemon L06-SecY-XX88XXX-q2.js"
  },
  "author": "[Your Name], Section Y, 6X88XXX",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.2",
    "nodemon": "^2.0.15"
  }
}
```

```
}
```

After setting up this script, you can run the node program `npm start` instead of `nodemon L06-SecY-XX88XXX-q2.js` (or `node L06-SecY-XX88XXX-q2.js`).

5. Download **q2.zip** consisting of two HTML files. Extract them in your directory.

```
L06-SecY-XX88XXX-q2
> node_modules
<> greeting_cn.html
<> greeting_th.html
|_ package-lock.json
|_ package.json
```

6. Create a file, **L06-SecY-XX88XXX-q2.js** The structure of your directory should be similar to

```
L06-SecY-XX88XXX-q2
> node_modules
<> greeting_cn.html
<> greeting_th.html
|_ L06-SecY-XX88XXX-q2.js
|_ package-lock.json
|_ package.json
```

Write the code to create a server that handles routing for each HTML page. **Note:** The server runs on port 3030

- a. `http://localhost:3030` → Show **"Hello World! in plain text"** (status:200)
- b. `http://localhost:3030/th` → Show `greeting_th.html` (status:200)
- c. `http://localhost:3030/cn` → Show `greeting_cn.html` (status:200)
- d. Otherwise, handle the invalid route → Show **"Where are you going?"** (status:404)

You can refer to the summary of the routing in the slide 54-55 or Learn more how to handle the unknown path [\[here\]](#) on "How does Express look up routes?" and "How to exploit the order of routes?" by Olusola Samuel

Here are the example of outputs from Q2 program:

- #1 Start your server with **npm start**

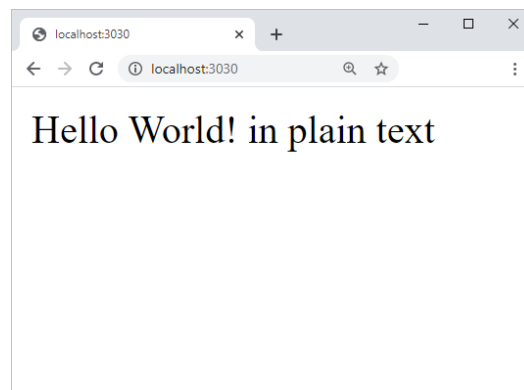
```
C:\[PATH]\L06-SecY-XX88XXX-q2>npm start
> l06-secy-xx88xxx-q2@1.0.0 start
> nodemon L06-SecY-XX88XXX-q2.js

[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node L06-SecY-XX88XXX-q2.js`
Running at Port 3030
—
```

#2 Type <http://localhost:3030/> in the browser

```
C:\[PATH]\L06-SecY-XX88XXX-q2>npm start
> l06-secy-xx88xxx-q2@1.0.0 start
> nodemon L06-SecY-XX88XXX-q2.js

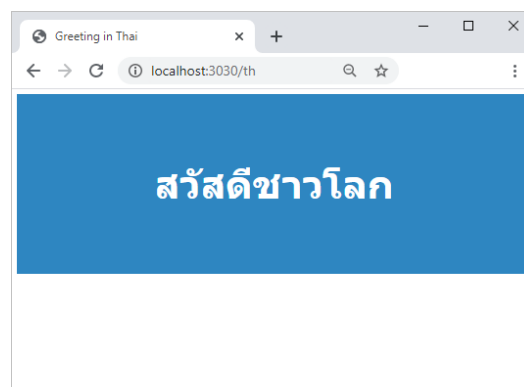
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node L06-SecY-XX88XXX-q2.js`
Running at Port 3030
—
```



#3 Type <http://localhost:3030/th> in the browser

```
C:\[PATH]\L06-SecY-XX88XXX-q2>npm start
> l06-secy-xx88xxx-q2@1.0.0 start
> nodemon L06-SecY-XX88XXX-q2.js

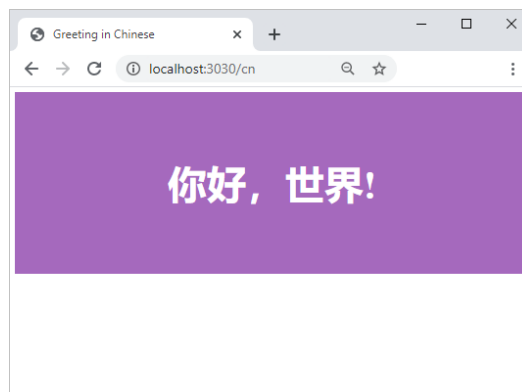
[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node L06-SecY-XX88XXX-q2.js`
Running at Port 3030
Req at: /
Req at: /th
—
```



#4 Type <http://localhost:3030/cn> in the browser

```
C:\[PATH]\L06-SecY-XX88XXX-q2>npm start
> l06-secy-xx88xxx-q2@1.0.0 start
> nodemon L06-SecY-XX88XXX-q2.js

[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node L06-SecY-XX88XXX-q2.js`
Running at Port 3030
Req at: /
Req at: /th
Req at: /cn
—
```

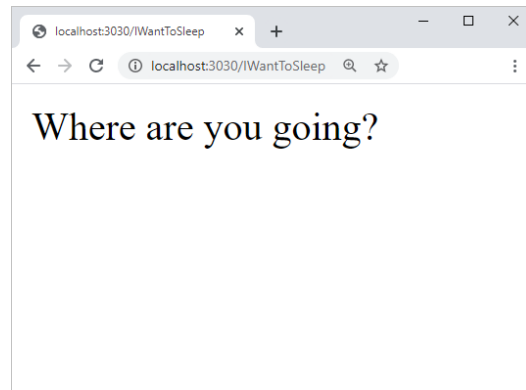


#5 Type the invalid url in the browser, e.g., <http://localhost:3030/IWantToSleep>.

Note: this should set the status of the response to 404 (error)

```
C:\[PATH]\L06-SecY-XX88XXX-q2>npm start
> l06-secy-xx88xxx-q2@1.0.0 start
> nodemon L06-SecY-XX88XXX-q2.js

[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node L06-SecY-XX88XXX-q2.js`
Running at Port 3030
Req at: /
Req at: /th
Req at: /cn
Req at: /IWantToSleep
—
```



Submit only the JS file, named as L06-SecY-XX88XXX-q2.js by class time via MyCourses (Note: XXX is your ID and Y is your section). **You must notify LAs to grade your work before submitting to MyCourses.**
