Node.jsProgramming_TimeServer

Use node.js to create a time server

the time server should run the current date & 24 hour time in the format:

"YYYY-MM-DD hh:mm"

Farnaz Golnam SFBU,MSCS#19576

Requirements:

JavaScript programming language is used for Node.js Application Development. The source files of Node.js applications have extension of ".js". Any text editor is sufficient to write Node.js code and save it as .js file.

- Install node.js
- Install npm (package manager for javascript) => cd your folder address > npm init => it will download package.json
- install browser sync: https://browsersync.io/ **=>** npm install --save browser-sync
- Install JavaScript ***date*** utility library => npm install date-fns --save

node-modules folder and package-lock.json file will be downloaded

Create a simple Node.js web server and handle HTTP requests:

To access web pages of any web application, you need a web server. The web server will handle all the http requests for the web application. Node.js provides capabilities to create your own web server which will handle HTTP requests asynchronously.

First, we import the http module using require() function. The http module is a core module of Node.js.

The next step is to call createServer() method of http and specify callback function with request and response parameter.

The http.createServer() method includes request and response parameters which is supplied by Node.js.

- The request object can be used to get information about the current HTTP request e.g., url, request header, and data.req.url is used to check the url of the current request and based on that it sends the response.To send a response:
  - first it sets the response header using writeHead() method
  - Then writes a string as a response body using write() method.
  - Finally, Node.js web server sends the response using end() method.
- The response object can be used to send a response for a current HTTP request.

Finally, call listen() method of server object which was returned from createServer() method with port number, to start listening to incoming requests on port 8000. You can specify any unused port here.

The following example is a simple Node.js web server contained in **time_server.js** file.

```
const http = require('http'); // Import Node.js core module
function zeroFill(i) {
   return (i < 10 ? '0' : '') + i
}
function now () {
   const d = new Date()
   return d.getFullYear() + '-'
      + zeroFill(d.getMonth() + 1) + '-'
      + zeroFill(d.getDate()) + ' '
      + zeroFill(d.getHours()) + ':'
      + zeroFill(d.getMinutes())
}
const server = http.createServer(function(req, res) { //create web server
   res.writeHead(200, {'Content-Type': 'text/plain'});
   res.end(now()+'\n');
});
server.listen(8000) //listen for any incoming requests
console.log('Node server is running on http://localhost:'+ 8000);
```

Run the above web server by writing node time_server.js command in command prompt or terminal window, ubuntu, etc and it will display message as shown below.

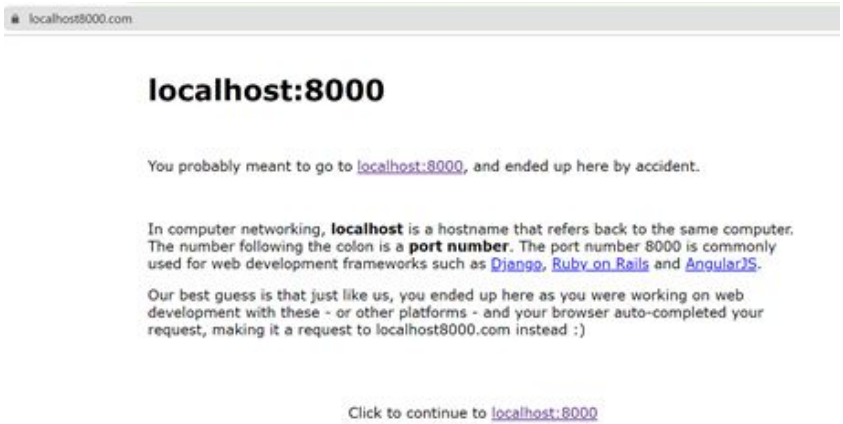
```
C:\Users\14152\Desktop\SFBU\4_Spring_2022\JavaScript\public_server>node time_server.js
Node server is running on http://localhost:8000
```

To test it

you can use the command-line program curl, which most Mac and Linux machines have pre-installed:

```
curl -i http://localhost:8000
```

For Windows users, point your browser to **http://localhost:8000** and see the following result.



Click on go to **localhost:8000** and type the full url address: