Node.js: Time Server

Step 1: Installing node js on ubuntu

Check ubuntu version and enable node repository

➔ Lsb_release -a

```
shagos90499@cs570:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.6 LTS
Release:        20.04
Codename:       focal
shagos90499@cs570:~$ curl -sL https://deb.nodesource.com/setup_19.x | sudo -E bash -

=============================================================================
=============================================================================
=============================================================================

                         SCRIPT DEPRECATION WARNING


  This script, located at https://deb.nodesource.com/setup_X, used to
  install Node.js is deprecated now and will eventually be made inactive.

  Please visit the NodeSource distributions Github and follow the
  instructions to migrate your repo.
  https://github.com/nodesource/distributions

  The NodeSource Node.js Linux distributions GitHub repository contains
  information about which versions of Node.js and which Linux distributions
  are supported and how to install it.
  https://github.com/nodesource/distributions


                         SCRIPT DEPRECATION WARNING

=============================================================================
=============================================================================
=============================================================================

TO AVOID THIS WAIT MIGRATE THE SCRIPT
Continuing in 60 seconds (press Ctrl-C to abort) ...
```

Installing nodejs and npm with command:

➔ sudo apt-get install -y nodejs

```
shagos90499@cs570:~$ sudo apt-get install -y nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 29.3 MB of archives.
After this operation, 189 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_19.x focal/main amd64 nodejs amd64 19.9.0-deb-1nodesource1 [29.3 MB]
Fetched 29.3 MB in 1s (40.9 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 77783 files and directories currently installed.)
Preparing to unpack .../nodejs_19.9.0-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (19.9.0-deb-1nodesource1) ...
Setting up nodejs (19.9.0-deb-1nodesource1) ...
Processing triggers for man-db (2.9.1-1) ...
shagos90499@cs570:~$
```

Check if NPM and node installed successfully:

- ➔ Node –version
- ➔ Npm --verison

```
shagos90499@cs570:~$ node --version
v19.9.0
shagos90499@cs570:~$ npm --version
9.6.3
shagos90499@cs570:~$
```

Step 2: create the node.js file to support the current time request to be displayed:

- ➔ Mkdir Time_server
- ➔ Cd Time_server

```
shagos90499@cs570:~$ mkdir Time_server
```

```
shagos90499@cs570:~$ ls
Pi_Calculations  Time_server  hadoop-3.4.0  hadoop-3.4.0.tar.gz
shagos90499@cs570:~$ cd Time_server/
shagos90499@cs570:~/Time_server$ vi http_json_api_time_server.js
```

Screenshot of Code: The function **currenttime** creates an object with the attributes of year, month, date, hour, and minute and returns the object which is then stringified with JSON format and displayed.

ssh.cloud.google.com/v2/ssh/projects/cs570-big-data-424809/zones/us-central1-f/insta...

ssh.cloud.google.com/v2/ssh/projects/cs570-big-data-424809/zones/us-central1-f/i...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
/////////////////////////////////////////////////////
//
// Write an HTTP server that serves JSON data when it
// receives a GET request to the path '/api/currenttime'.
//
/////////////////////////////////////////////////////
var http = require('http')
// Function to make an object with year, month, date, hour,
// minute format and return the object
function currenttime() {
    const now = new Date();

    // Formatting the date and time
    const dateTime = {
        year: now.getFullYear(),
        month: String(now.getMonth() + 1).padStart(2, '0'),
        date: String(now.getDate()).padStart(2, '0'),
        hour: String(now.getHours()).padStart(2, '0'),
        minute: String(now.getMinutes()).padStart(2, '0')
    };

    return dateTime;
}

var server = http.createServer(function (req, res) {
  // req.url = /api/currenttime
  var result
  // match req.url with the string /api/currenttime
  if (/^\/api\/currenttime/.test(req.url))
    result = currenttime()

  if (result) {
    res.writeHead(200, { 'Content-Type': 'application/json' })
    res.end(JSON.stringify(result) + '\n\n')
  } else {
    res.writeHead(404)
    res.end('Not Found.\n')
  }
})
server.listen(Number(process.argv[2]), () => {
    console.log('Server is running on port 8000');
});
-- INSERT --                                                    1,1          All
```

Step 3: Then run the server by providing a third argument as the port number:

Terminal 1:

➔ node http_json_api_time_server.js 8000

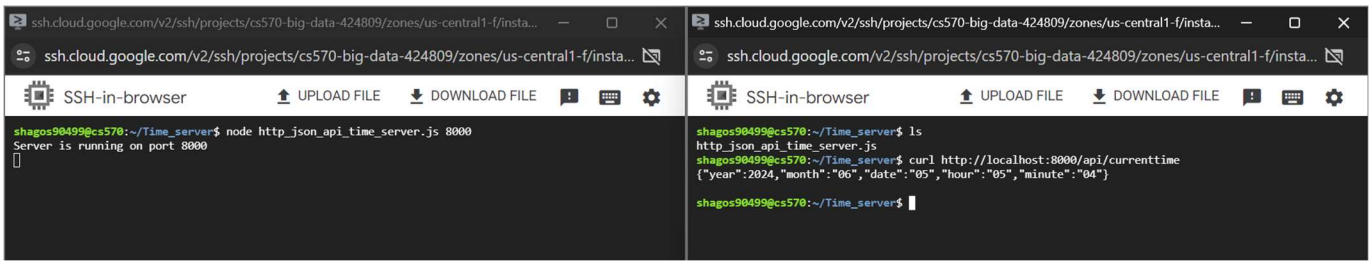
```
shagos90499@cs570:~/Time_server$ node http_json_api_time_server.js 8000
Server is running on port 8000
```

You can see that the server is running on 8000, next the output of the result can be displayed by using curl.

To send a GET request to the server and see the result, “curl” can be used. So, open another terminal, and run the following command to send a GET request. This allows you to simulate a web browser or other HTTP client by making a request to the Node.js server. “Curl” retrieves and displays the JSON output returned by the server directly in your terminal.

Terminal 2:

➔ curl http://localhost:8000/api/currenttime



As you can see, the server is running on port 8000, and on the second terminal on the right, the JSON output is retrieved displaying time and date.

Get output like this:

{"year":2024,"month":"06","date":"05","hour":"08","minute":"06"}

```
shagos90499@cs570:~/Time_server$ curl http://localhost:8000/api/currenttime
{"year":2024,"month":"06","date":"05","hour":"05","minute":"19"}
```