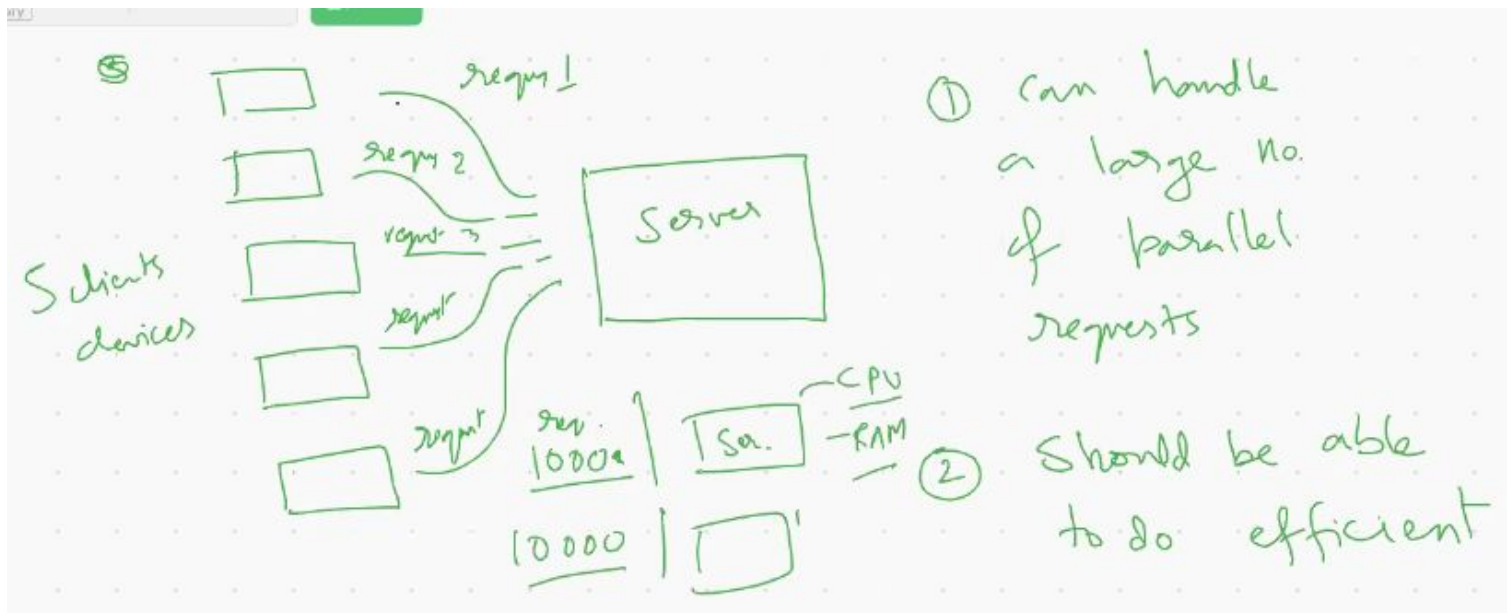


Day: 9.3

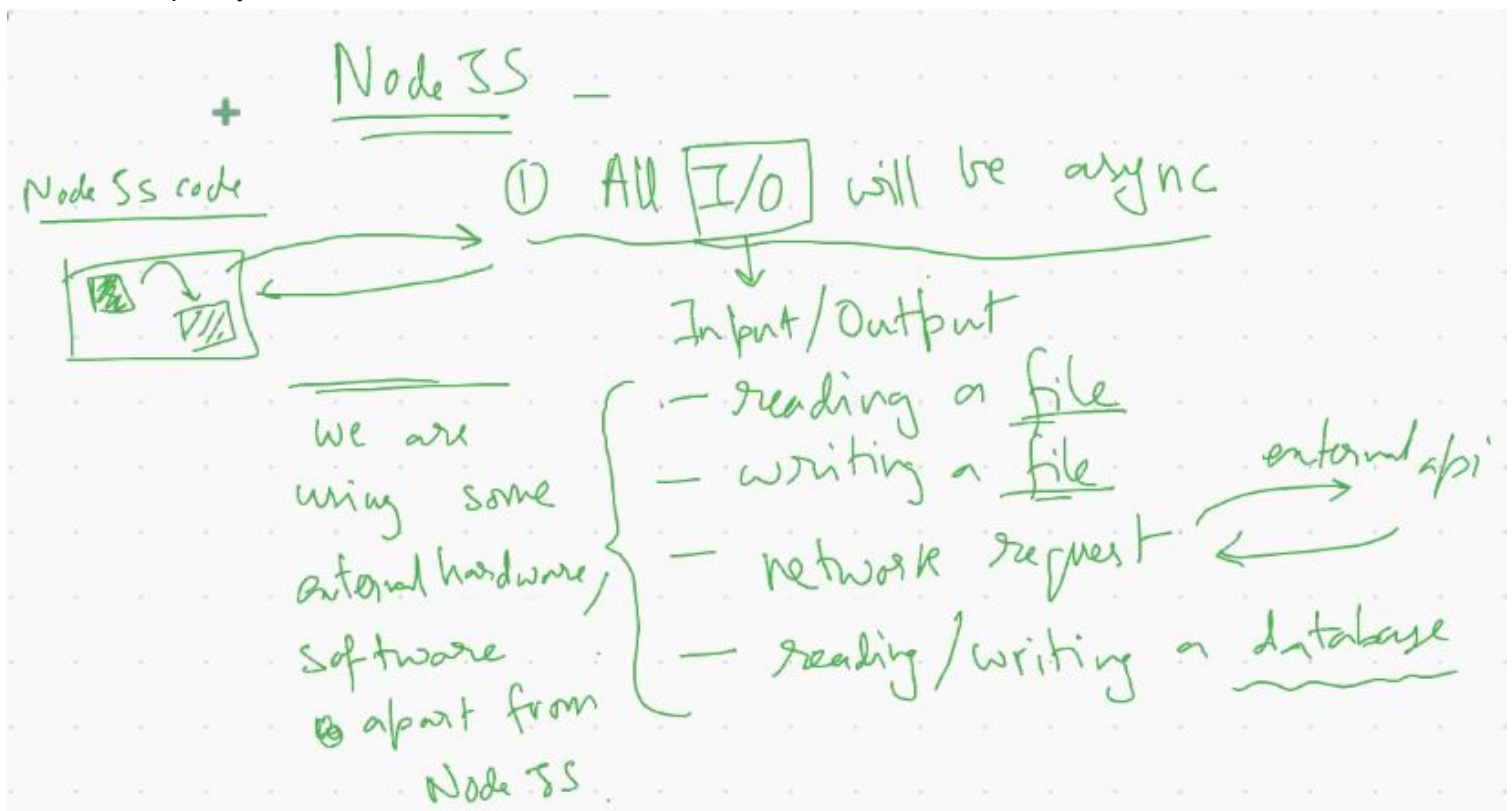
Title: Node: Setting up first server - 25th November, 2020

Topics:



Java is multithreaded

Javascript is single threaded, how javascript is handling this many request on a single thread, it does it because of the event loop 1 by 1,



• newton school leaderbord

api → call this api (JSON body)

① Understand the request
— (JSON parsing)

• newton school leaderbord

api → call this api (JSON body)

① Understand the request
— (JSON parsing) / authentication check

I/O ② gets the response data from database/file

③ create the proper response object [nammer]

conversion to appropriate type say JSON

I/O ④ return the response to the client

Node JS

① All I/O will be async

Input/Output

- reading a file
- writing a file
- network request ← external api
- reading/writing database

JS obj ≠ JSON

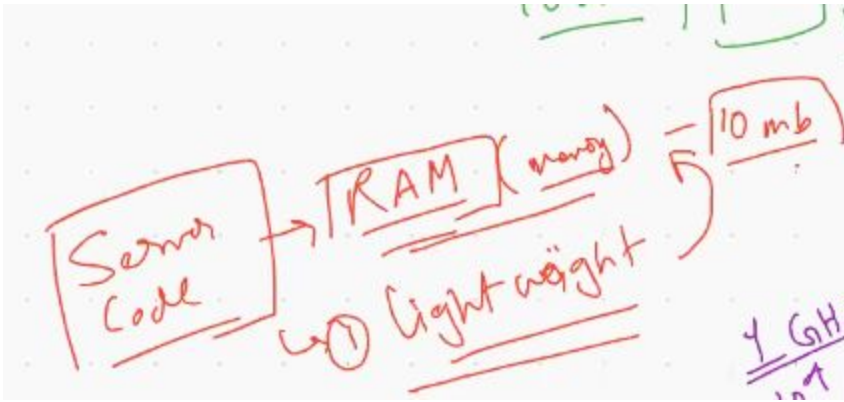
← JSON.stringify()

Node JS code

we are using some external hardware

software part from Node JS

Throttling is the similar term



Server should take less amount of ram around 10mb and it should be light weight
Code of server should take less then 10mb of ram

Nodejs is efficient because by default it has very less global items, whatever is needed by nodejs we have to explicitly import, nodejs does not have such backage like browser has like alert confirm etc

Things that are loaded in nodejs
By default it loads

- ① set Timeout / set Interval
- ② console.log
- ③ Promise, Date, Math
- ④ process

By default any i/p operation is not loaded by default

Install this application before moving forward

- ① node / npm / vs code
- ② Postman

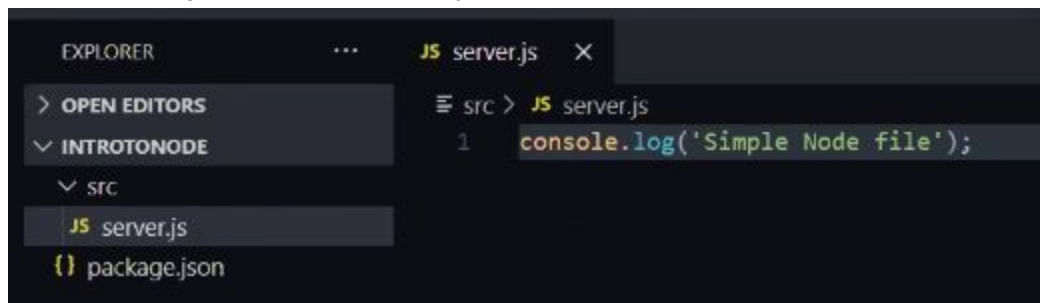
For creating nodejs project

```
C:\Users\Debanshu\Documents\Newton School\IntroToNode>npm init
```

Enter enter default things
Package.json is created

Create a directory called src

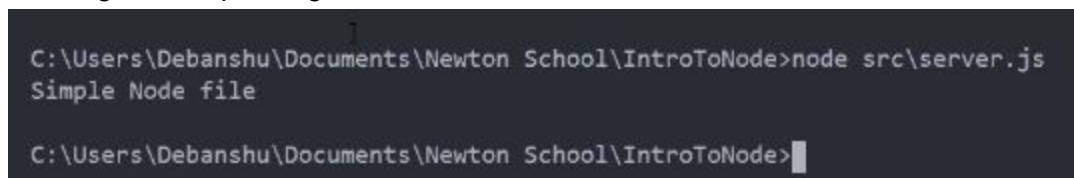
Create server.js inside this directory



The image shows the VS Code interface. The Explorer sidebar on the left shows a project named 'INTROTONODE' with a subdirectory 'src'. Inside 'src', there is a file 'server.js' and a file 'package.json'. The Editor pane shows the content of 'server.js' with the following code:

```
src > JS server.js
1 console.log('Simple Node file');
```

Running this script using node



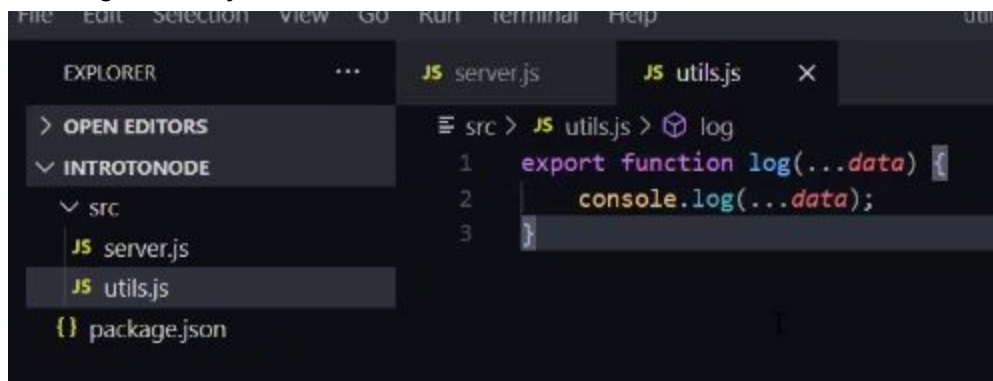
The image shows a terminal window with the following commands and output:

```
C:\Users\Debanshu\Documents\Newton School\IntroToNode>node src\server.js
Simple Node file

C:\Users\Debanshu\Documents\Newton School\IntroToNode>
```

Assume you have external library-> utils.js

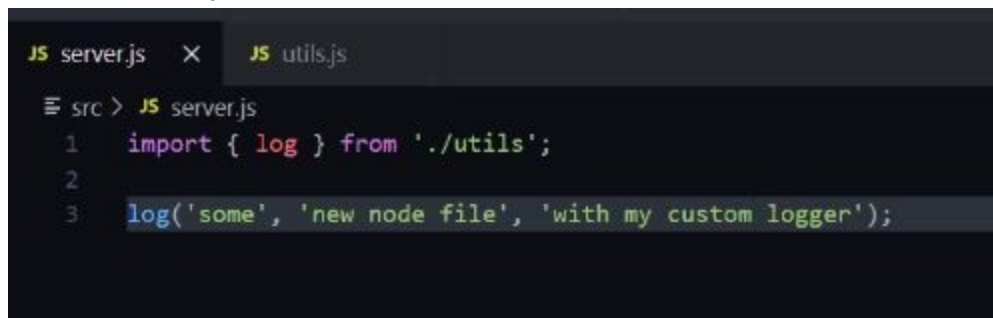
Creating file utils.js and function to it



The image shows the VS Code interface. The Explorer sidebar on the left shows the 'src' directory with files 'server.js', 'utils.js', and 'package.json'. The Editor pane shows the content of 'utils.js' with the following code:

```
src > JS utils.js > log
1 export function log(...data) {
2   console.log(...data);
3 }
```

Modified server.js



The image shows the VS Code Editor with the 'server.js' file open. The code is as follows:

```
src > JS server.js
1 import { log } from './utils';
2
3 log('some', 'new node file', 'with my custom logger');
```

```

C:\Users\Debanshu\Documents\Newton School\IntroToNode>node src\server.js
Simple Node file

C:\Users\Debanshu\Documents\Newton School\IntroToNode>node src\server.js
C:\Users\Debanshu\Documents\Newton School\IntroToNode\src\server.js:1
import { log } from './utils';
^^^^^^

SyntaxError: Cannot use import statement outside a module
    at wrapSafe (internal/modules/cjs/loader.js:1072:16)
    at Module._compile (internal/modules/cjs/loader.js:1122:27)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:1178:10)
    at Module.load (internal/modules/cjs/loader.js:1002:32)
    at Function.Module._load (internal/modules/cjs/loader.js:901:14)
    at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:74:12)
    at internal/main/run_main_module.js:18:47

C:\Users\Debanshu\Documents\Newton School\IntroToNode>

```

Error because of version 12 of node

```

C:\Users\Debanshu\Documents\Newton School\IntroToNode>node -v
v12.16.1

```

Do non es6 import and export

```

JS server.js  X  JS utils.js
src > JS server.js > ...
1  const { log } = require("./utils");
2
3  log("some", "new node file", "with my custom logger");
4

```

```

JS server.js  JS utils.js  X
src > JS utils.js > ...
1  function log(...data) {
2    |    console.log(...data);
3  }
4
5  module.exports.log = log;

```

```

C:\Users\Debanshu\Documents\Newton School\IntroToNode>node src\server.js
some new node file with my custom logger

C:\Users\Debanshu\Documents\Newton School\IntroToNode>

```

Non default exports are also known as named exports

Example of named export module.exports.log

```

C:\Users\Debanshu\Documents\Newton School\IntroToNode>node src\server.js
some new node file with my custom logger

C:\Users\Debanshu\Documents\Newton School\IntroToNode>node src\server.js
some new node file with my custom logger

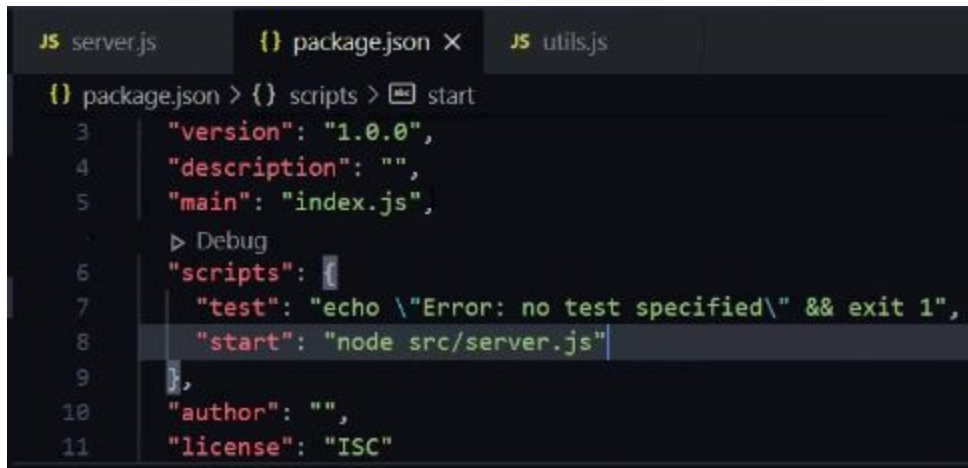
C:\Users\Debanshu\Documents\Newton School\IntroToNode>npm start
npm ERR! missing script: start

npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\Debanshu\AppData\Roaming\npm-cache\_logs\2020-11-25T17_22_14_865Z-debug.log

C:\Users\Debanshu\Documents\Newton School\IntroToNode>

```

Package.json modified



```

JS server.js  {} package.json X  JS utils.js
{} package.json > {} scripts > start
3  "version": "1.0.0",
4  "description": "",
5  "main": "index.js",
6  "scripts": {
7    "test": "echo \"Error: no test specified\" && exit 1",
8    "start": "node src/server.js"
9  },
10 "author": "",
11 "license": "ISC"

```

```

C:\Users\Debanshu\Documents\Newton School\IntroToNode>npm start

> introtonode@1.0.0 start C:\Users\Debanshu\Documents\Newton School\IntroToNode
> node src/server.js

some new node file with my custom logger

C:\Users\Debanshu\Documents\Newton School\IntroToNode>

```

Importing http module

By default nodejs doesn't have anything to create server, so that's why we have to import something ,
 Http is the inbuilt module of node
 and this allows us to write servers

```
JS server.js  {} package.json  JS utils.js
src > JS server.js > [e] server > http.createServer() callback
1  const { log } = require("./utils");
2  const http = require('http');
3
4  const server = http.createServer((req, res) => {
5    log('Received request', req);
6    res.write('Hello Newton School');
7    res.end();
8  })
9
```

Imported http module and created server using it

createServer() function takes two properties

First is the request and second is the response

Request is where you can read the request

Response is where you can write something to the response

res.end() ends the response

And last thing is to give a port number

```
JS server.js  X  {} package.json  JS utils.js
src > JS server.js > ...
1  const { log } = require("./utils");
2  const http = require('http');
3
4  const server = http.createServer((req, res) => {
5    log('Received request', req);
6    res.write('Hello Newton School');
7    res.end();
8  });
9
10 server.listen(9999);
11
```

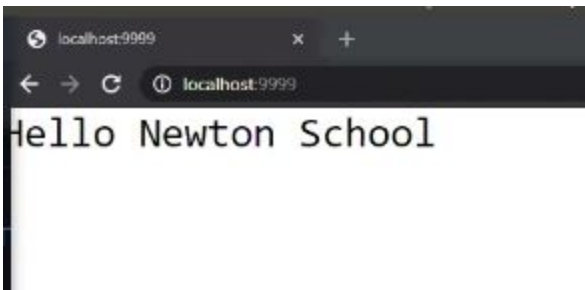
That's it

We wrote a very simple server

Always use port number above 8000

```
TERMINAL  ...  1: node  +  [ ]  [ ]  ^  x
C:\Users\Debanshu\Documents\Newton School\IntroToNode>npm start
> introtonode@1.0.0 start C:\Users\Debanshu\Documents\Newton School\IntroToNode
> node src/server.js
[ ]
```

SERVER STARTED



And in terminal you can see so many things, so server tells that there was a request came on this server

```
> node src/server.js  
  
Received request /  
Received request /favicon.ico  
█
```

started server again and receive request printed when reloaded the browser

page this one



Installing postman on ubuntu

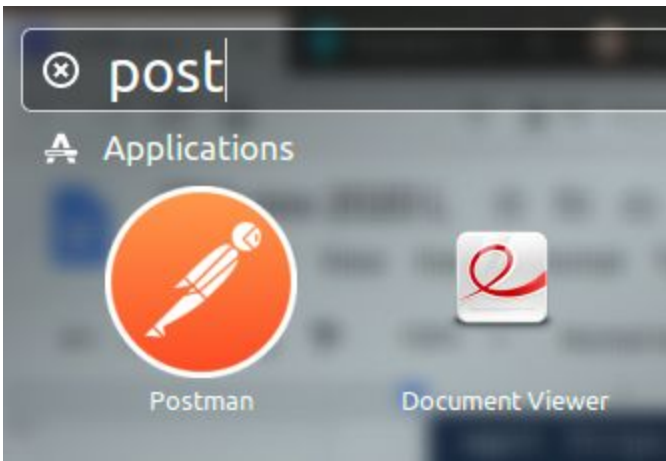
```
wget https://dl.pstmn.io/download/latest/linux64 -O postman.tar.gz  
sudo tar -xzf postman.tar.gz -C /opt  
rm postman.tar.gz  
sudo ln -s /opt/Postman/Postman /usr/bin/postman
```

For adding it to unity desktop

```
cat > ~/.local/share/applications/postman.desktop <<EOL  
[Desktop Entry]  
Encoding=UTF-8  
Name=Postman  
Exec=postman  
Icon=/opt/Postman/app/resources/app/assets/icon.png  
Terminal=false  
Type=Application  
Categories=Development;  
EOL
```

Now log off your system and login again

Hit the windows button and search for postman



Code

```
wget https://dl.pstmn.io/download/latest/linux64 -O postman.tar.gz
sudo tar -xzf postman.tar.gz -C /opt
rm postman.tar.gz
sudo ln -s /opt/Postman/Postman /usr/bin/postman
```

```
cat > ~/.local/share/applications/postman.desktop <<EOL
[Desktop Entry]
Encoding=UTF-8
Name=Postman
Exec=postman
Icon=/opt/Postman/app/resources/app/assets/icon.png
Terminal=false
Type=Application
Categories=Development;
EOL
```

Reference:

<https://www.bluematador.com/blog/postman-how-to-install-on-ubuntu-1604>

End of the doc