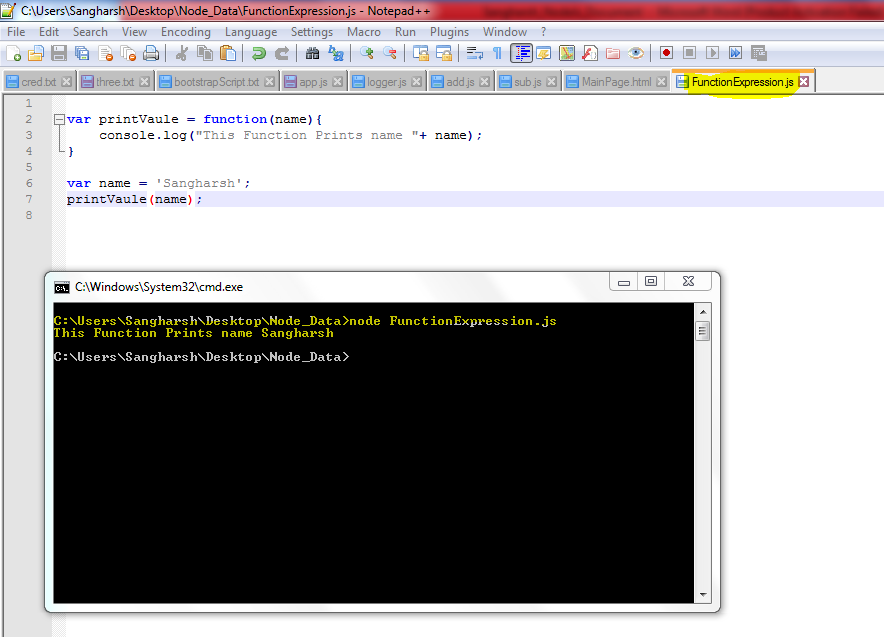
**NodeJs:**

**Global Objects:**

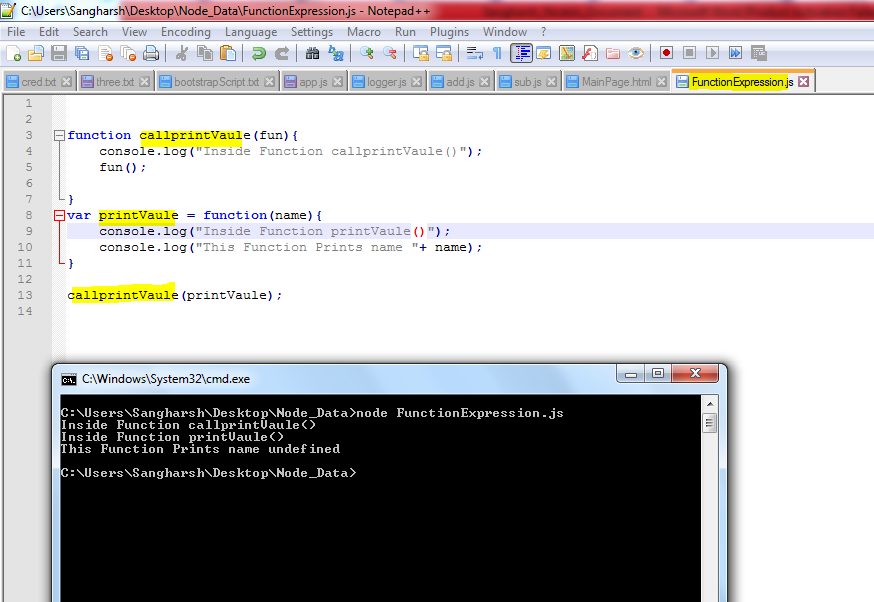
* console
* module
* \_\_dirname
* \_\_filename
* Require()

**Function Expressions:**

* The main difference between a function expression and a function declaration is the function name.
* A function expression can used as an IIFE(Immediately Invoked Function Expression) which runs as soon as it is defined.



* In Nodejs One function can call other function

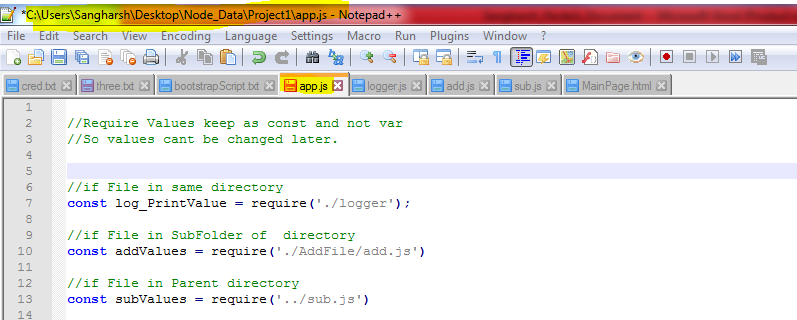


**Module and require ()**

* **module** and not **modules**
* module can be **inbuilt(**provided by node **ex. events ,require , response , file system )** or **custom(written by us)**
* individual files in nodejs are termed as **module**
* **module.exports** and not **modules.export** or **module.export**
* **Using require () to get files from Different or same Directories.**

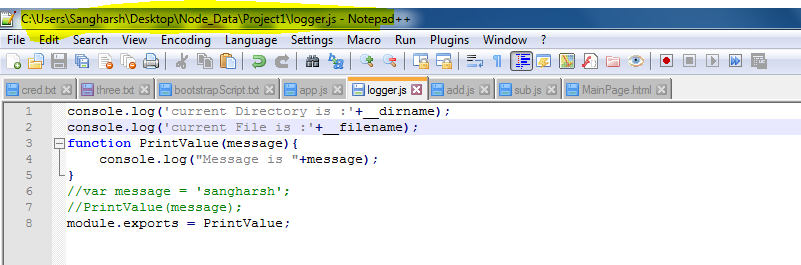
Below File Code Represents how using require we can import different files.

Path: **C:\Users\Sangharsh\Desktop\Node\_Data\Project1\app.js**

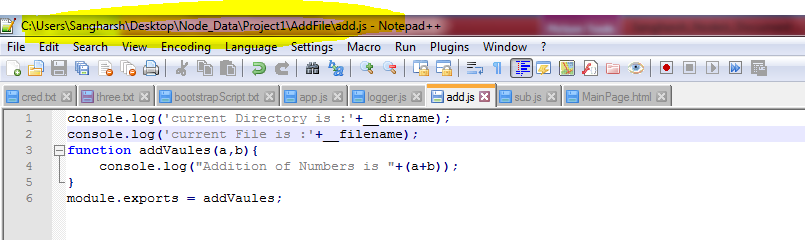


* module.exports return values.

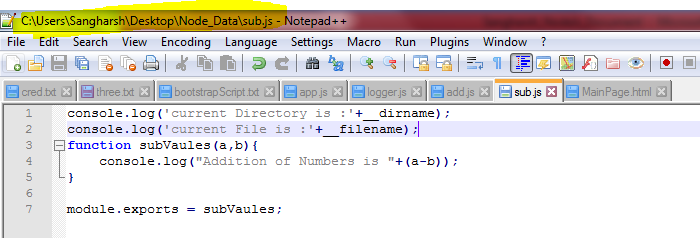
Path: **C:\Users\Sangharsh\Desktop\Node\_Data\Project1\logger.js**



Path: **C:\Users\Sangharsh\Desktop\Node\_Data\Project1\AddFile\add.js**

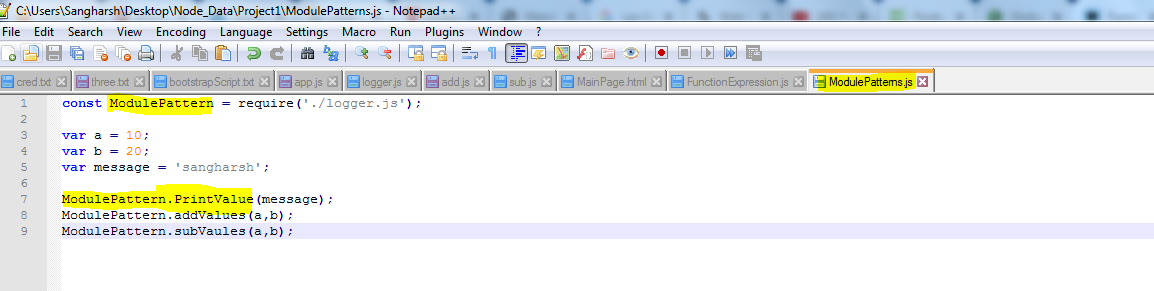


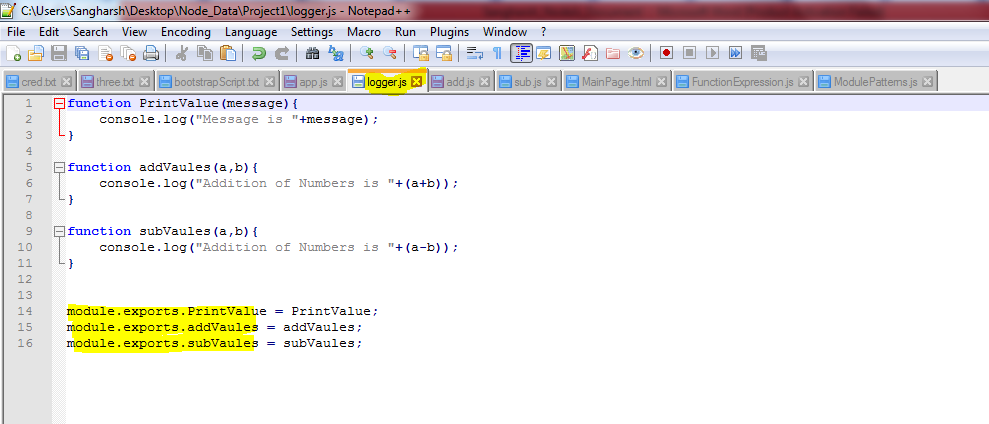
PATH: **C:\Users\Sangharsh\Desktop\Node\_Data\sub.js**



**Module Patterns:**

* Data can be exported individually.





**Events Module:**

All objects that emit events are instances of the EventEmitter class.

const EventEmitter = require('events');

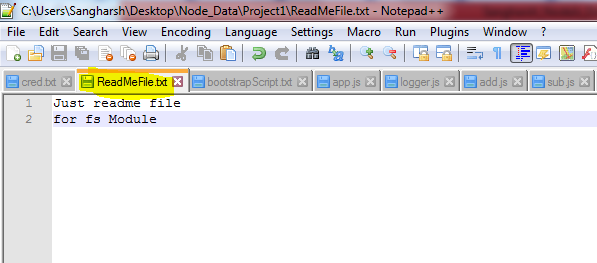
class MyEmitter extends EventEmitter {}

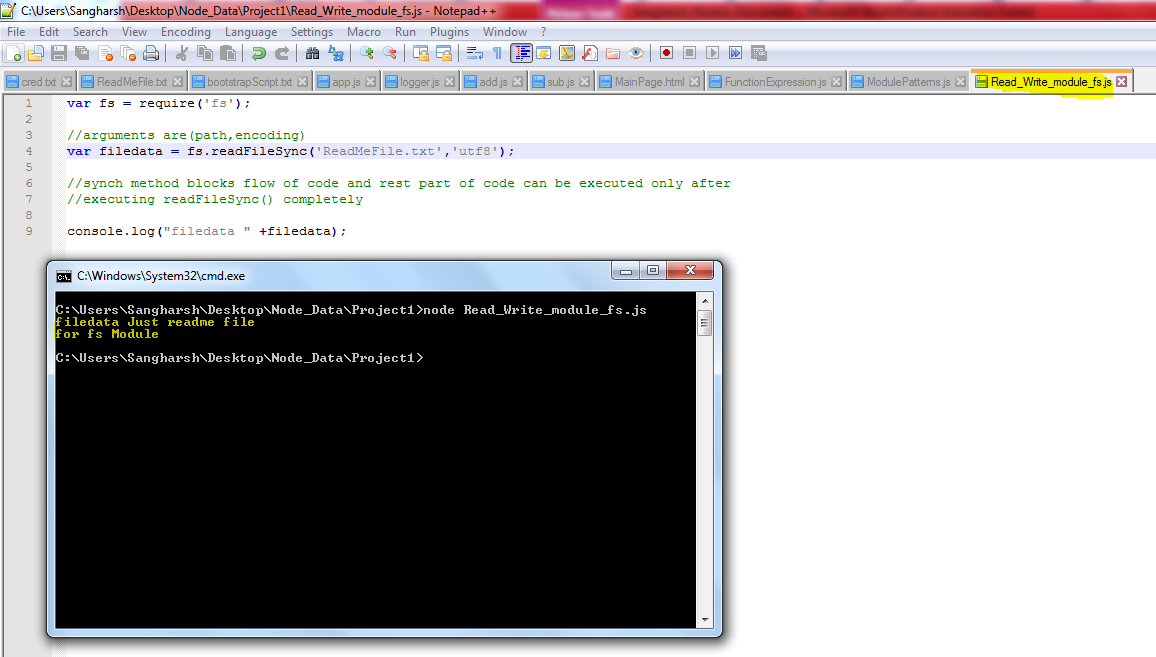
const myEmitter = new MyEmitter();

EventEmitter is a class.

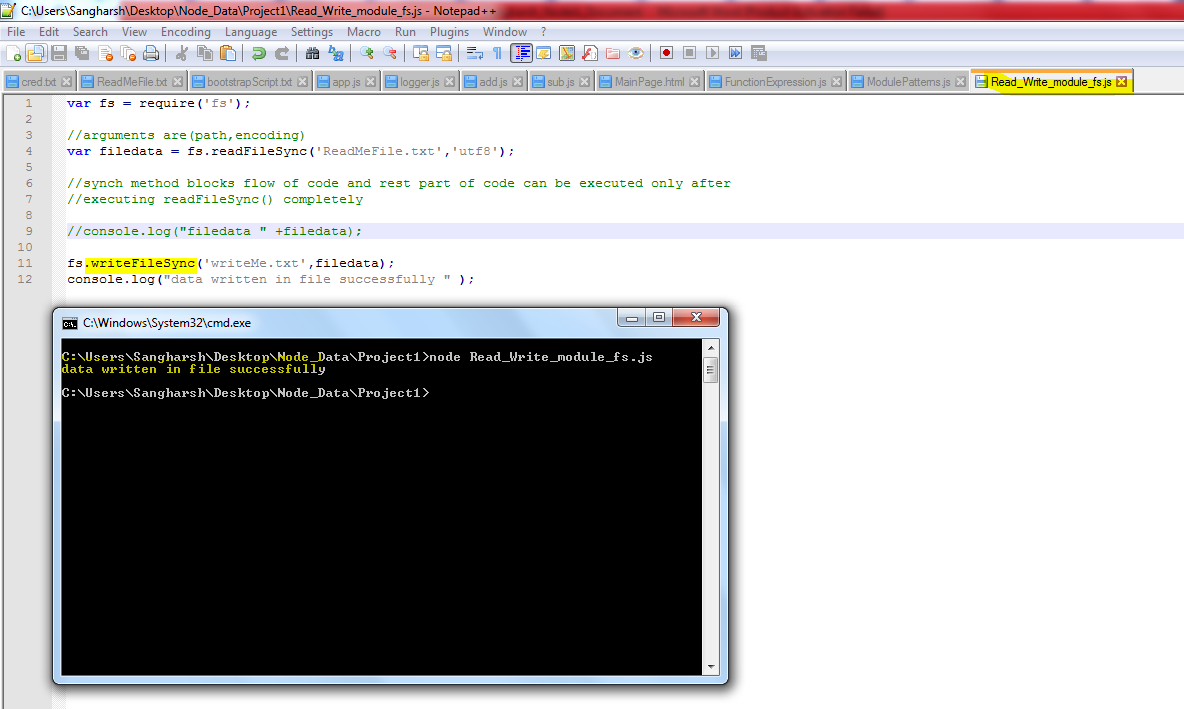
**Reading and Writing in Files:**

* **Module required is ‘fs’**
* **Reading data from file:**
* **synch method blocks flow of code and rest part of code can be executed only after**
* **readFileSync() is function used**

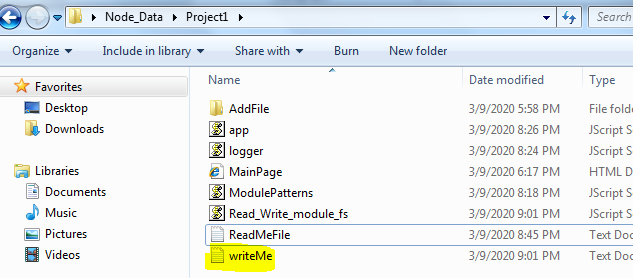


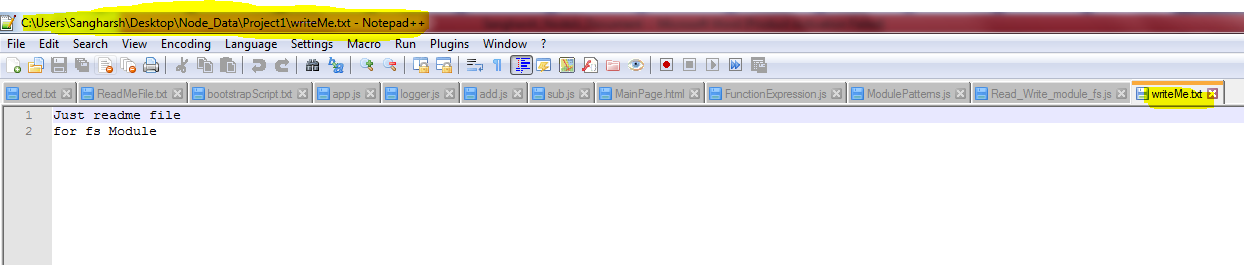


* **Writing data in file:**
* **writeFileSync () is function used**

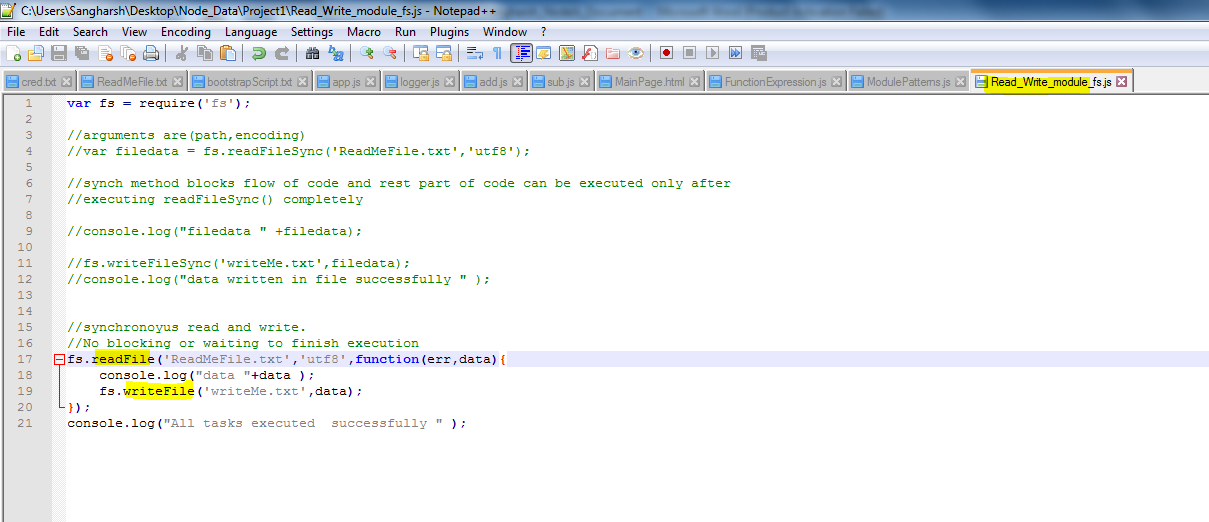


**File generated successfully.**





* For asynchronous Read and write
* No blocking or waiting to finish execution



**Add/Remove Directories both synchronously and asynchronously**.

* mkdir / rmdir
* mkdirSync / rmdirSync

**NOTE:**

* While using asynchronous methods use callback functions to do something once action is completed.
* **Remove** **files** in directory then only you can **delete** **directory**.

**Clients and Servers:**

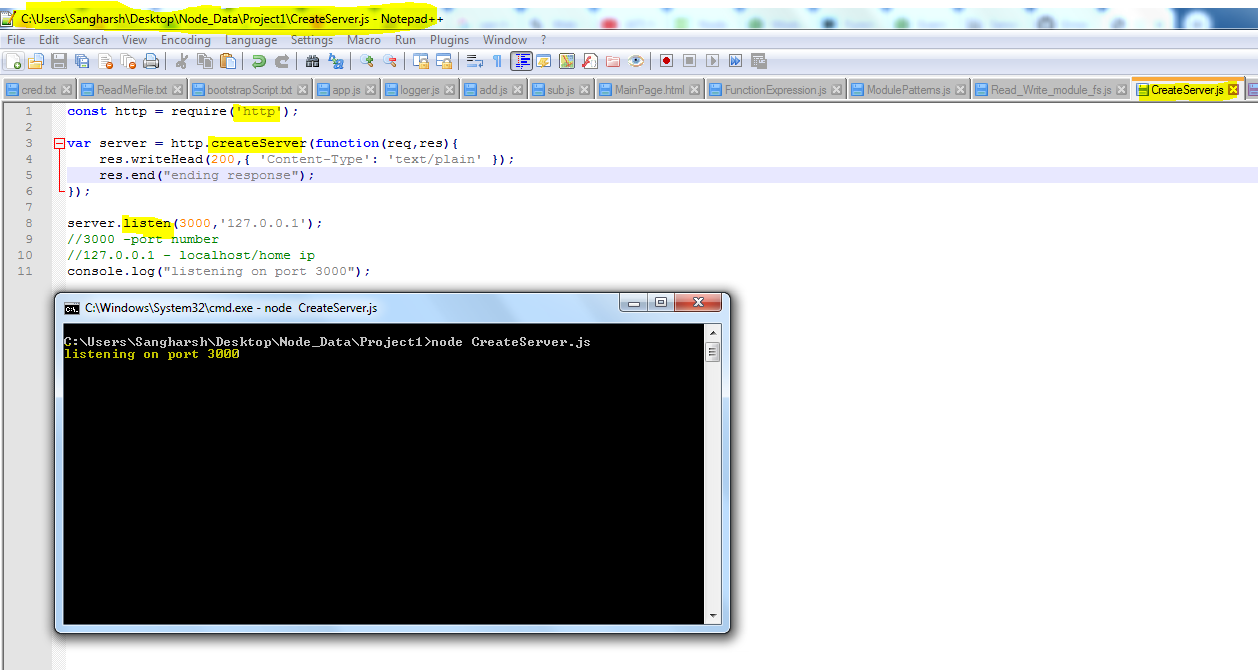
* **Client** makes **Request** and **Server** Sends **Response using some Protocols (Set of well-defined Rules)**.

**Creating a Server:**

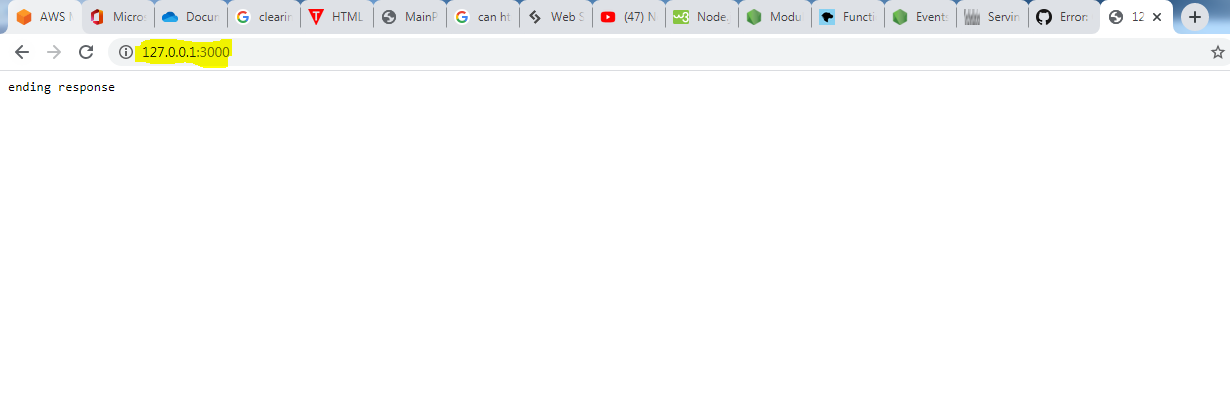
* Module used is ‘**http’**

**var server = http.createServer();**

* Status 200 = ok
* Status 404 = page not found.
* http.createServer()
* server.listen()



* Listening on port 3000

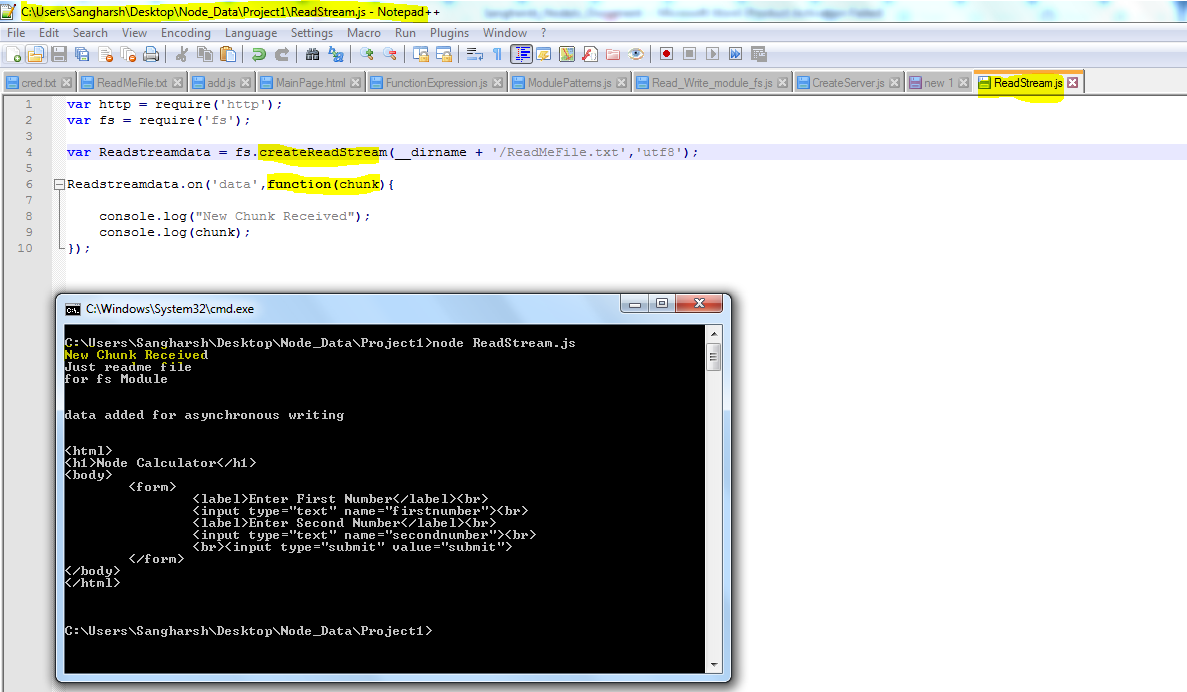


**Streams and Buffers:**

* Buffers are temporary storage spots of data
* Data transferred bit by bit in **streams**(writable and Readable).

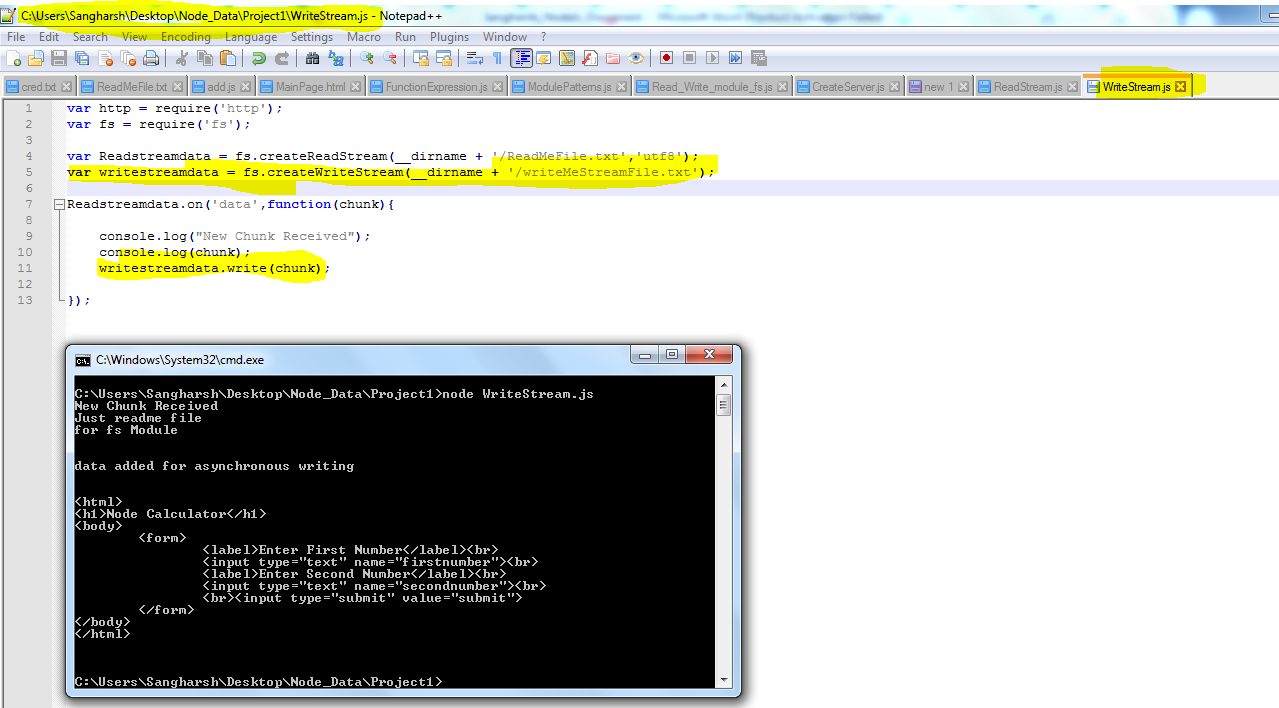
Creating Read Stream:

* Using fs.**createReadStream**(\_\_dirname + '/ReadMeFile.txt','utf8');
* Reading is done in **pieces** called **chunk**

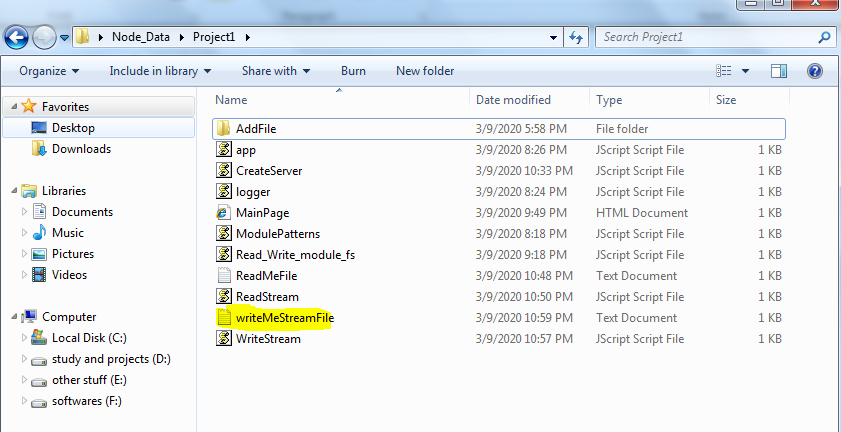


Creating write Stream:

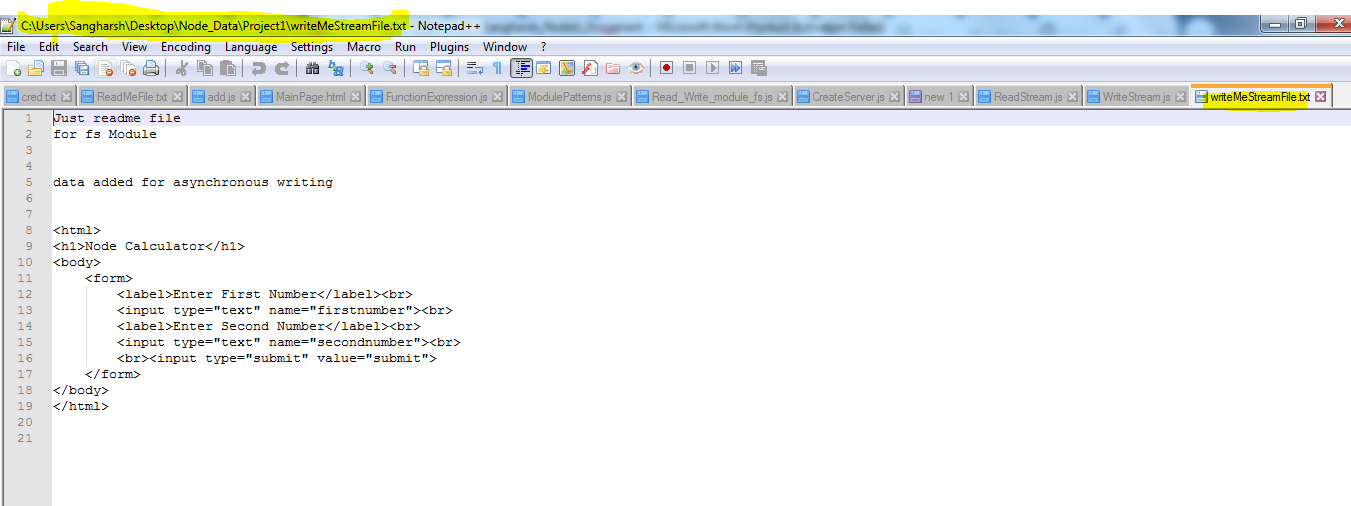
* Using fs.**createWriteStream**(\_\_dirname + '/Name of file to write');
* writing is done in **pieces** called **chunk**



File created successfully



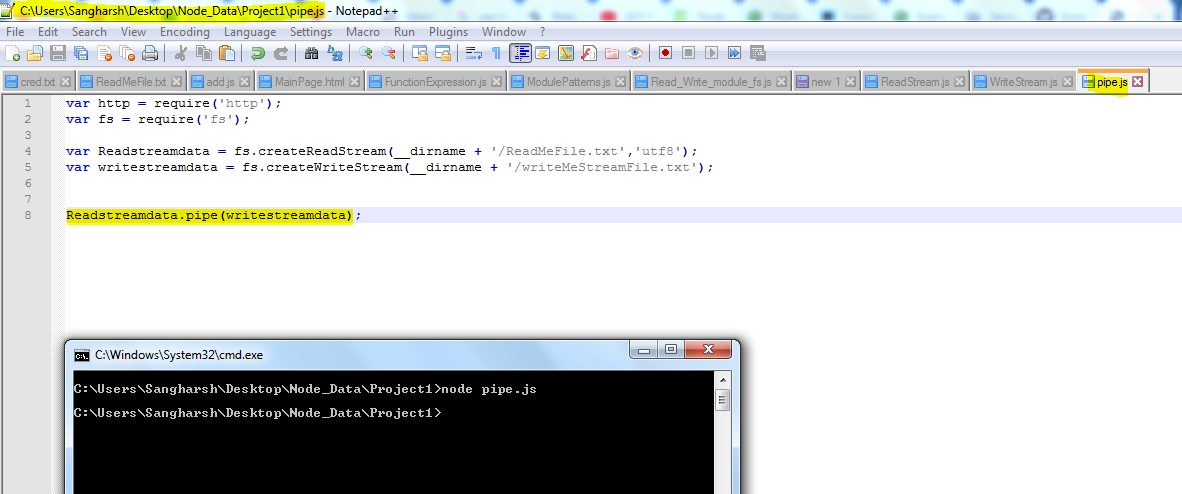
Data copied



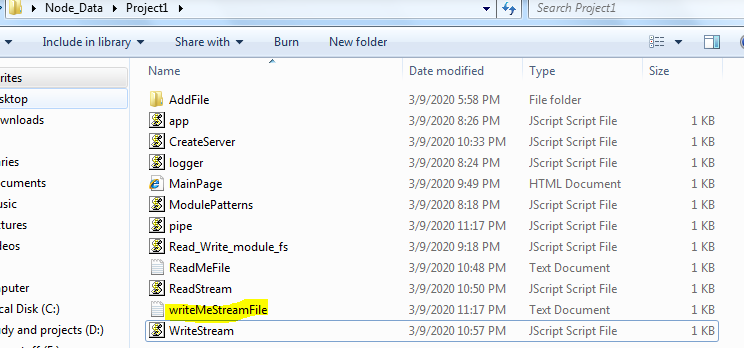
* Data read from Read Stream and written into Write Stream in small chunks.

**Pipes:**

* No need of manual listening to read data streams by using pipes
* But we need ReadStream and writeStream for piping data.

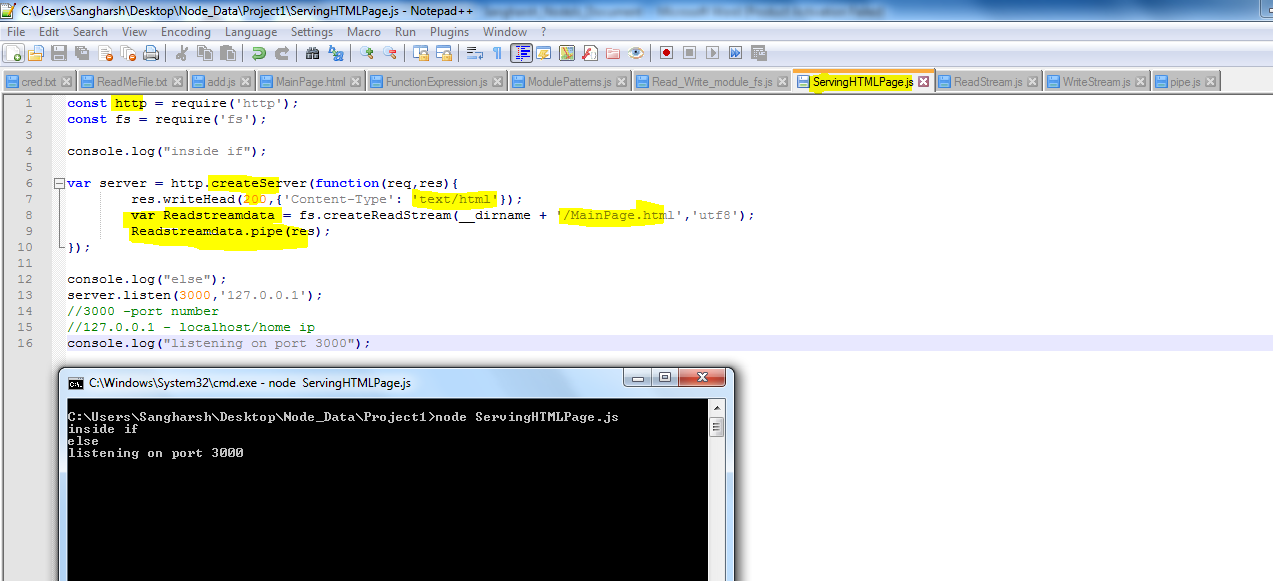


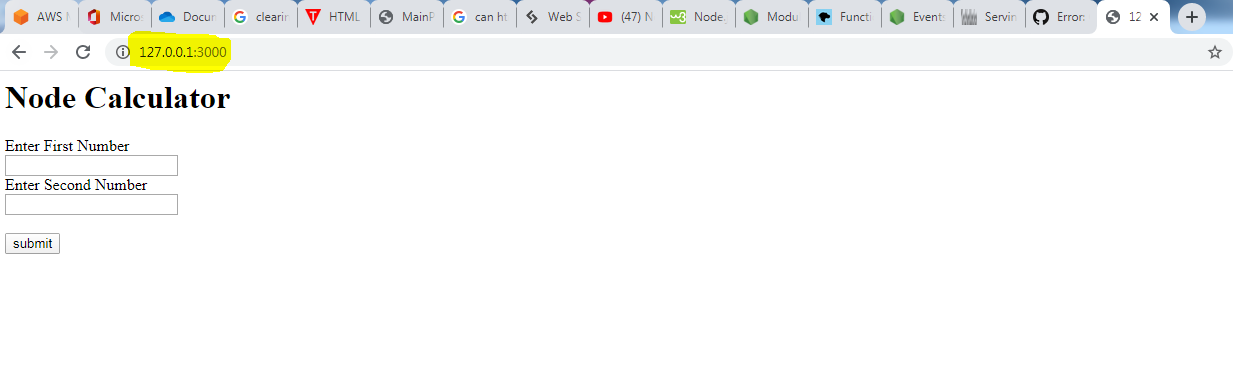
File created and data copied



**Serving HTML Pages:**

* **res.writeHead**(200,{'Content-Type': **'text/html'**});
* **Read** data from **html page** in **ReadStream** and then using **pipes** send it to **response(browser)**





**Serving JSON Data:**

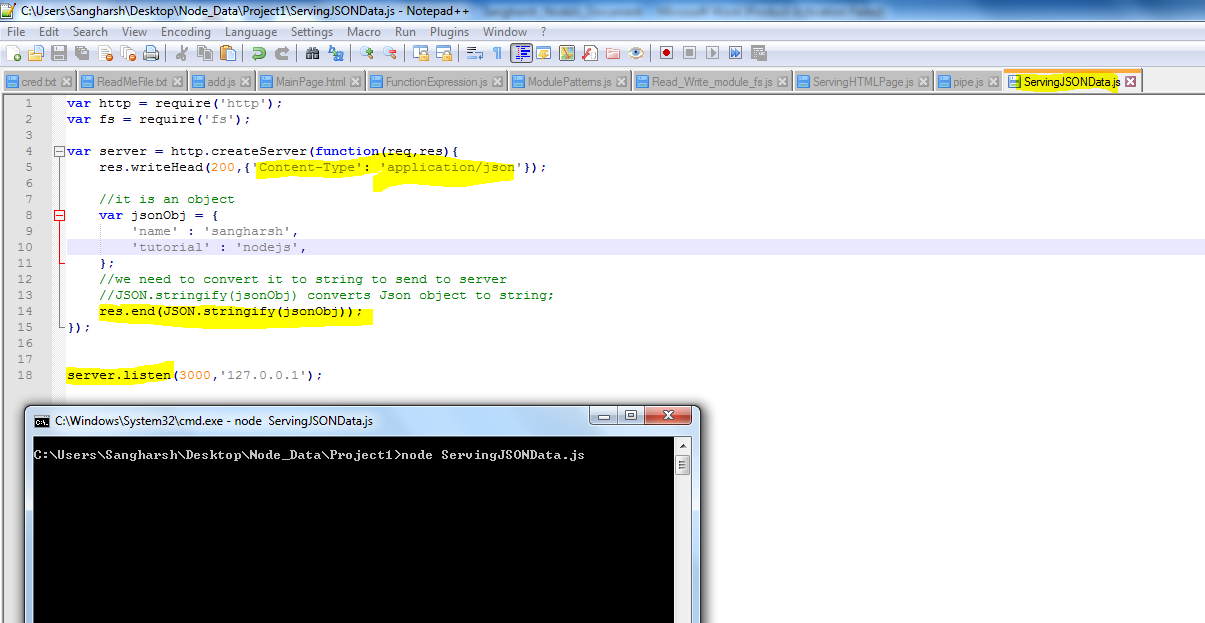
**var jsonObj = {**

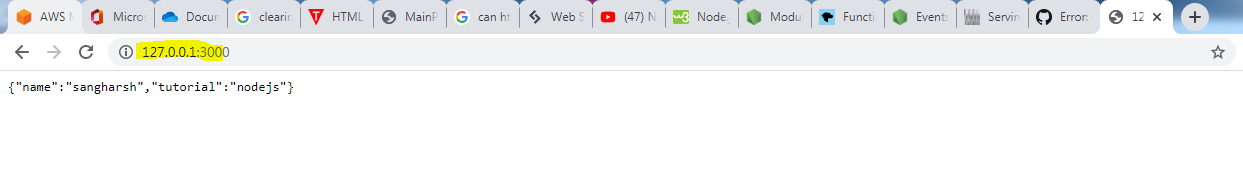
**'name' : 'sangharsh',**

**'tutorial' : 'nodejs',**

**};**

* we need to convert it to string to send to server
* JSON.**stringify**(jsonObj) converts Json object to string



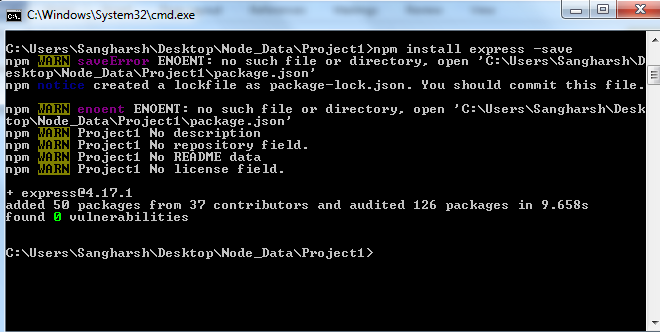


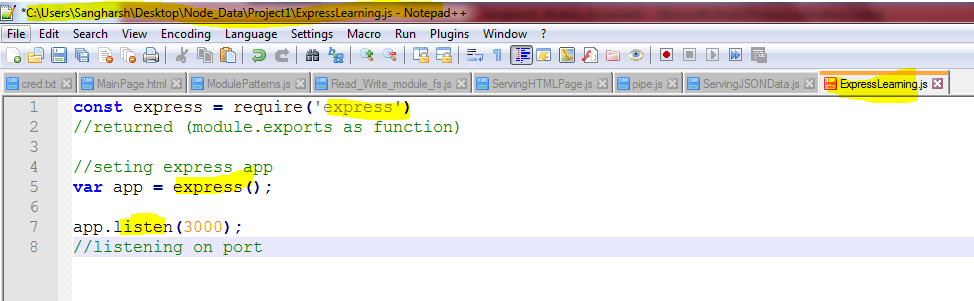
**Express:**

* Important Node module
* Routing is Easy

**To install express**

* **npm install express –save**



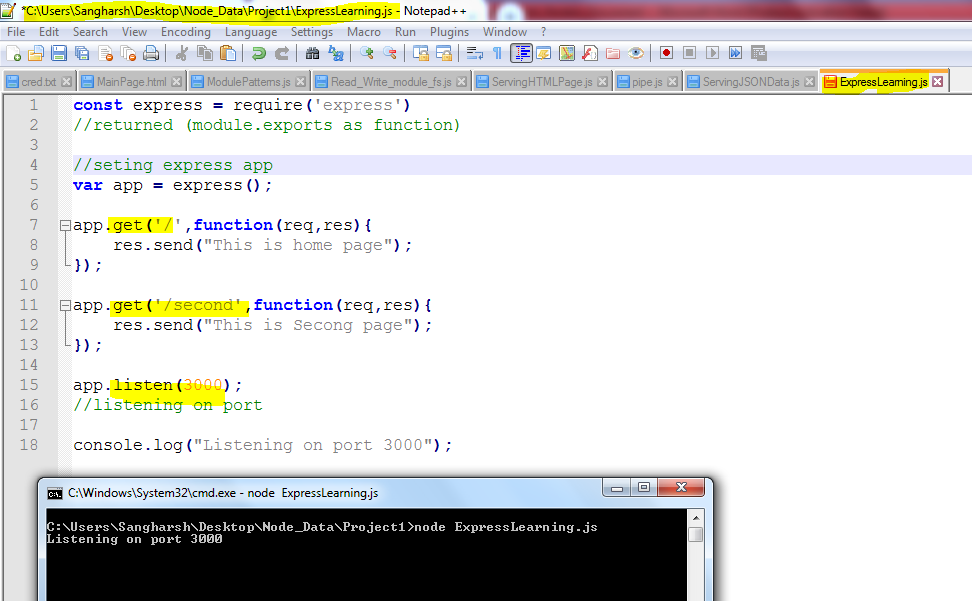


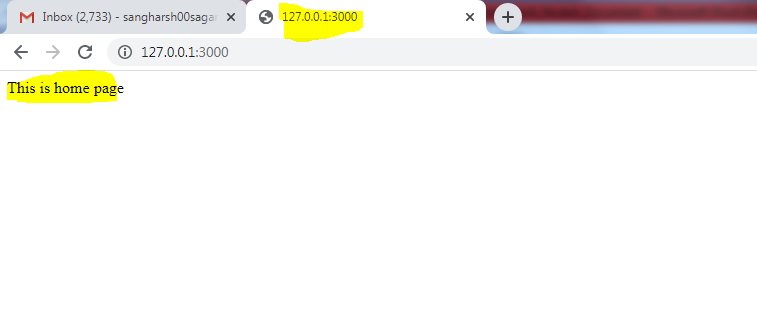
* **Var express = require(‘express’);**
* **Var app = express();**

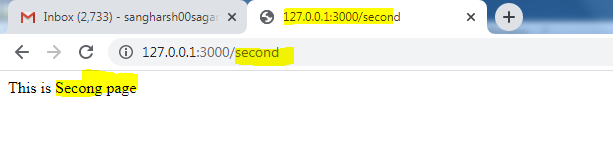
**HTML Methods:**

* GET ----- app.get(‘route’,function)
* PUT ----- app.put(‘route’,function)
* POST ----- app.post(‘route’,function)
* DELETE----- app.delete(‘route’,function)

**Routing using app.get()**







**Route Parameters in express:**

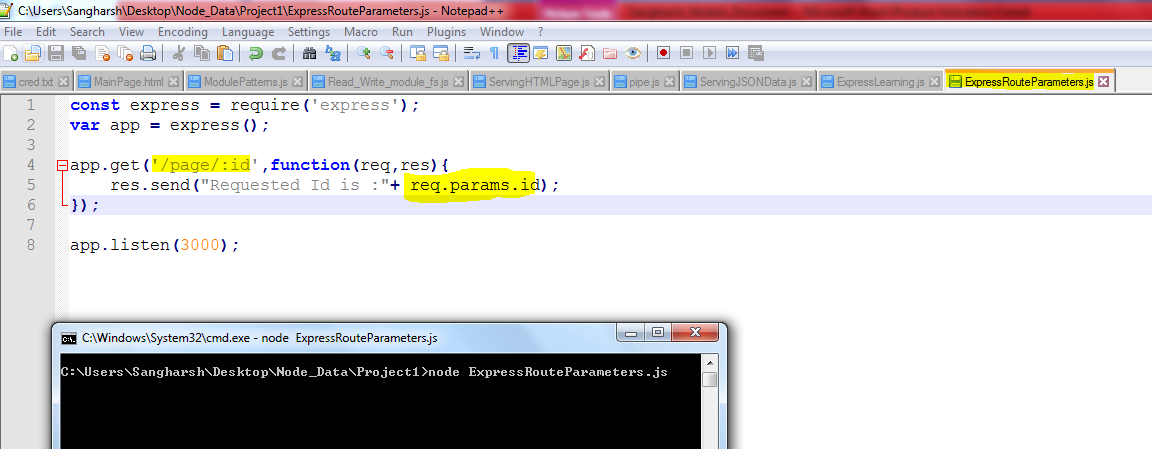
* Setting Dynamic Routes

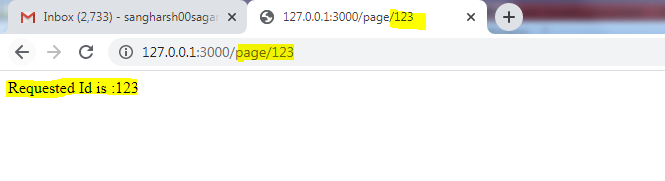
app.get('/page/:**id'**,function(req,res){

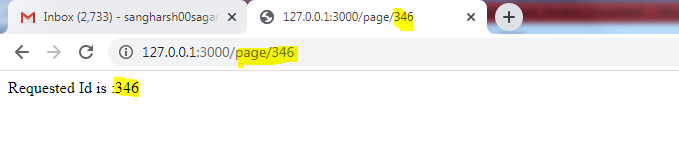
res.send("Requested Id is :"+ **req.params.id**);

});

* Here **Id** is dynamic in nature you can change it to get different values.







**Template Engines:**