Lab 3 (individual) Node JS Module, Debug, backtick, GTE server date and time, file system

Hide Folder Information

Instructions

A. debugging
B. a simple GET request server-rendered app to which the user sends requests from constructing the query strings at the browser address bar
C. Write and read files remotely

Note! you cannot use nodejs express for this lab or any lab

A(2 mark) debugging.jpg

The purpose of this part is to demonstrate that debugging is functioning properly on your system (follow the instructions attached to this lab, or simple google how to install nodejs)

In Visual studio code (preferred IDE) create a module, math.js, with two functions to add and subtract two numbers.

call the math module functions in your code, app.js, and place a breakpoint (to practice debugging)

Place a breakpoint in your code, with console.log statement like the one in the image below

note:

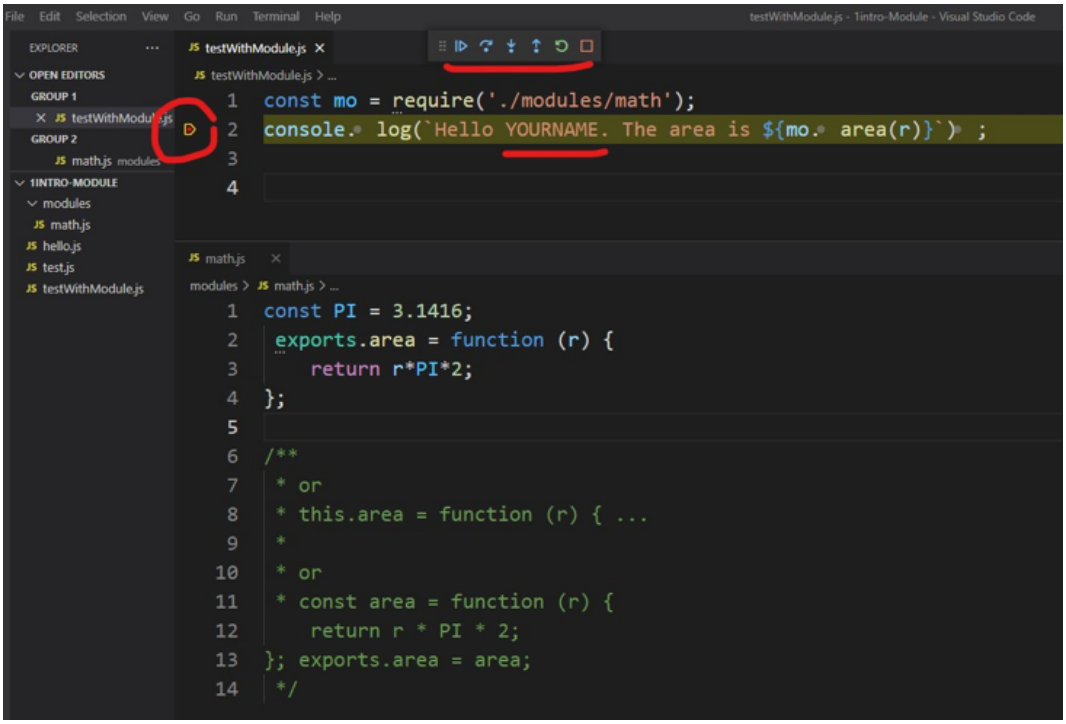
1- Use backtick format

2- replace YOURNAME with your first name (so that you don't accidently submit somebody else's screen shot)

start debugging and make screenshot proving that the breakpoint pauses execution of your script, just like in the attached pdf file)

for part A you only need to submit the screenshot to prove that the debugging works

and the breakpoint hits! e.g.



B (6 marks) simple API call to get the time of server

In part B you implement your very first API server (well a GET server) . So far, you have been a consumer of third party APIs (such as google APIs, Weather API etc), now you are going to take your very first step to make your own.

Develop a server side nodejs service to get your name you sent via the browser's address bar, greet you and return the current time of the server (entire message in blue)*. You need to host it remotely**

* the message the server returns has to contain the in-line styling to be displayed in blue. Don't style the message in the client side.

**again as stated in lecture notes and the learning hub, it is your responsibility to obtain a web hosting service to host your assignment (refer to course outline and learning hub home page).

In your browser you want to send your name to the server from the address bar of the browser (which makes it a ? call).

... that's right a GET call.

Example:

an API request like this one at the browser's address bar

https://yourDomainName.xyz/COMP4537/labs/3/getDate/?name=John**

Shall return this message in **blue**:

Hello John, What a beautiful day. Server current date and time is * Wed Sept 01 2023 12:52:14 GMT-0800 (Pacific Standard Time)

*: whatever current date and time of the **server** is. So it *has to come from the server, not from your browser.*

**Note: For the endpoint of your API url (server side), you can choose another path (url) that you find more appropriate, it has to be https though

Your part B project must be composed of two files:

1- the **server.js**

2- **./modules/utils.js** for the definition of your function getDate() {...

(you need to put your function in a module)

Note: As you can see, there is no need to have any client side implementation, your web app is a **server-rendered** one

Store the user facing string (Hello %1, What a beautiful day. Server current date and time is) in a separate file

lang/en/en.js or en.json or anything you prefer

or

locals/en/...

C(2 marks) Writing to a file, Reading from a file

C.1

https://yourDomainName.xyz/COMP4537/labs/3/**writeFile**/?text=BCIT*

which appends a new line with text BCIT to the text file file.txt

You need to append the text (which includes in the url) to the exiting file (file.txt). Do not re-create the file every time the user sends the request, just append to the exiting one.
Append means:
if the file does not exist, create it and then store the text to it
if the file exists already, just add (append) the text of query string to the bottom of the existing file.

C.2

https://yourDomainName.xyz/COMP4537/labs/3/**readFile**/file.txt*

which reads the entire content of the updated file file.txt and returns it back to the browser and displays it at the browser's page (it must not download the file, it has to display its content)

if the file does not exist, your app has to return 404 error message including the file name the user had entered

*__Note__: For the endpoint of your API url (server side), you can chose another path (url) that you find more appropriate

Deliverable:

A(file): yourLastNameDebuggingLab3.jpg
Upload the **screenshot of part A** and the urls of part B (1 url) and part C (2 urls) , e.g.

together with your urls of part B (1 url) and (part c, **2** urls) with https at comments section of Learning hub
e.g.
B(url)- **https**://yourDomainName.xyz/COMP4537/labs/3/getDate/?name=John (***replace John with your name***)

C(urls)
C.1- **https**://yourDomainName.xyz/COMP4537/labs/3/writeFile/?text=BCIT
C.2- **https**://yourDomainName.xyz/COMP4537/labs/3/readFile/file.txt
At the time of marking, we try them with various the url query strings (we change John and BCIT to see the result accordingly)

File to upload (part B together with part C if done)

zip: **YourLastNameLab3.zip**
zip your Lab 3 files from the root directory and put them all in one zip file: yourLastNameLab3.zip e.g. GreenLab3.zip

Rubrics (additional deductions)

(-6) if Part B or C is not hosted

(-4) if the url(s) are not posted at the comment sections (which is as equal as not hosting at all)

(-2) if you post the urls at the comment section but forgot to include https:// and make it hyperlinked
(-2) if making the message in blue color is not done or is on the client side

(-1) if you dont store the user facing message (Hello %1, What a beautiful day. Server current date and time is) in a separate file

(-2) of not fully OOP (everything inside classes)

-10% mark deduction for each day late, 0 after three days late submission

Due on Sep 29, 2024 11:59 PM

Available on Sep 24, 2024 8:30 AM. **Access restricted before availability starts.**

Attachments

Install-debug-nodeJs-Visual-Studio-Code.pdf (248.03 KB)

node.js_example.zip (1.59 KB)

Download All Files

Submit Assignment

Files to submit

(0) file(s) to submit

After uploading, you must click Submit to complete the submission.

Add a File **Record Audio** **Record Video**

Comments

Submit **Cancel**