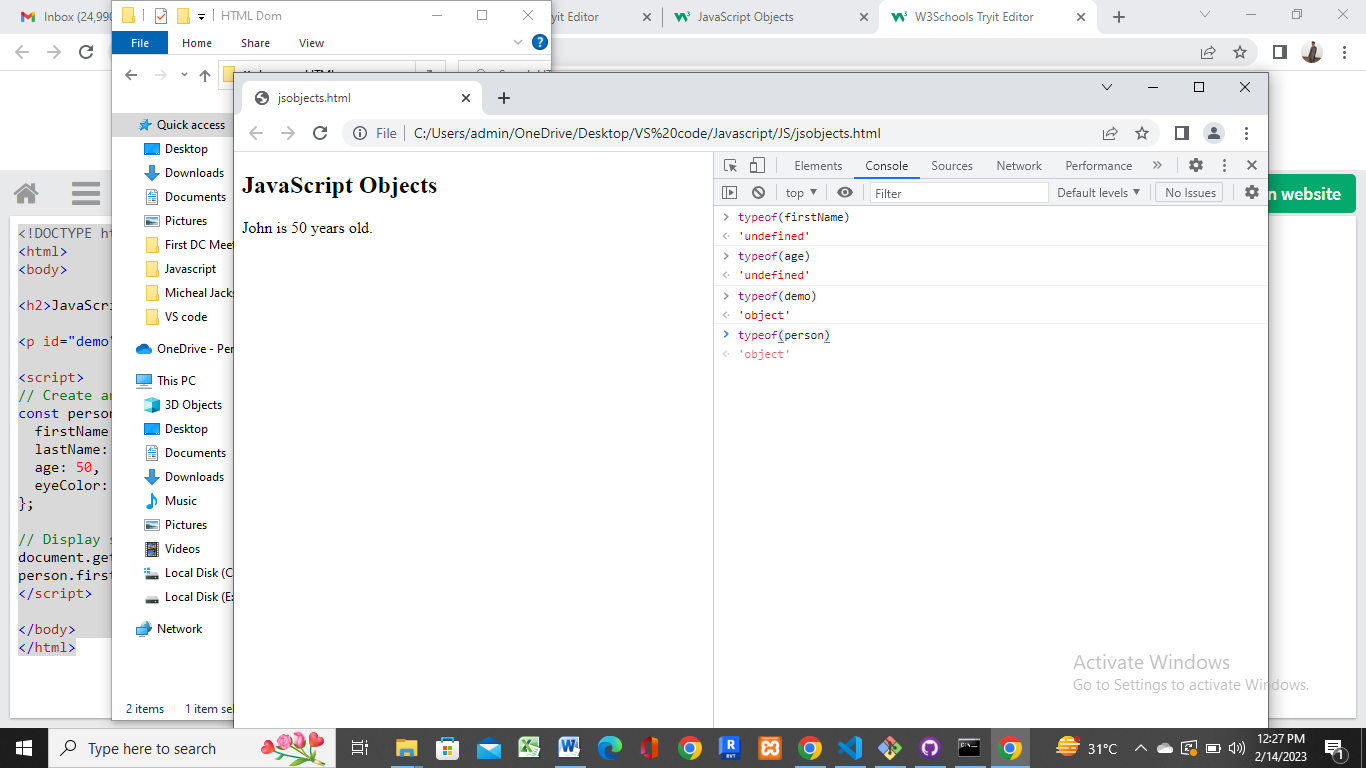
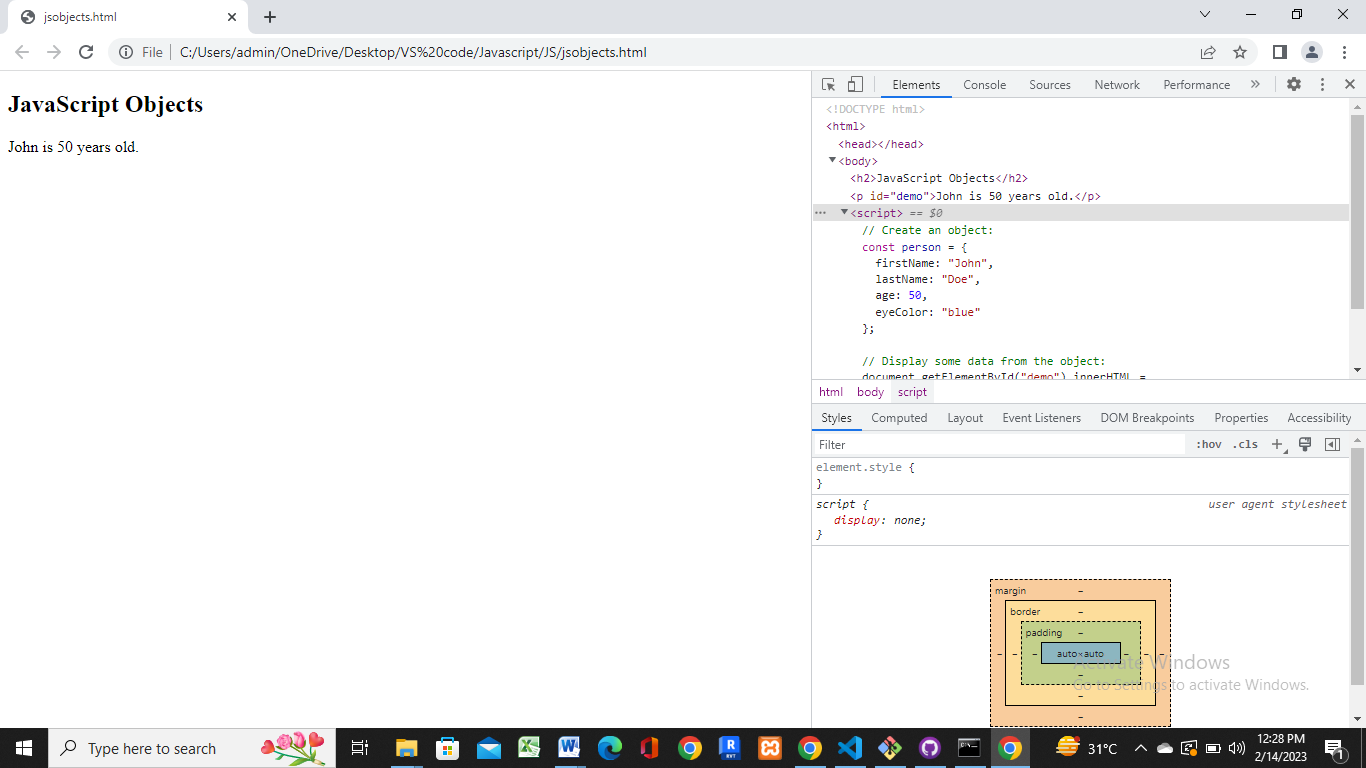
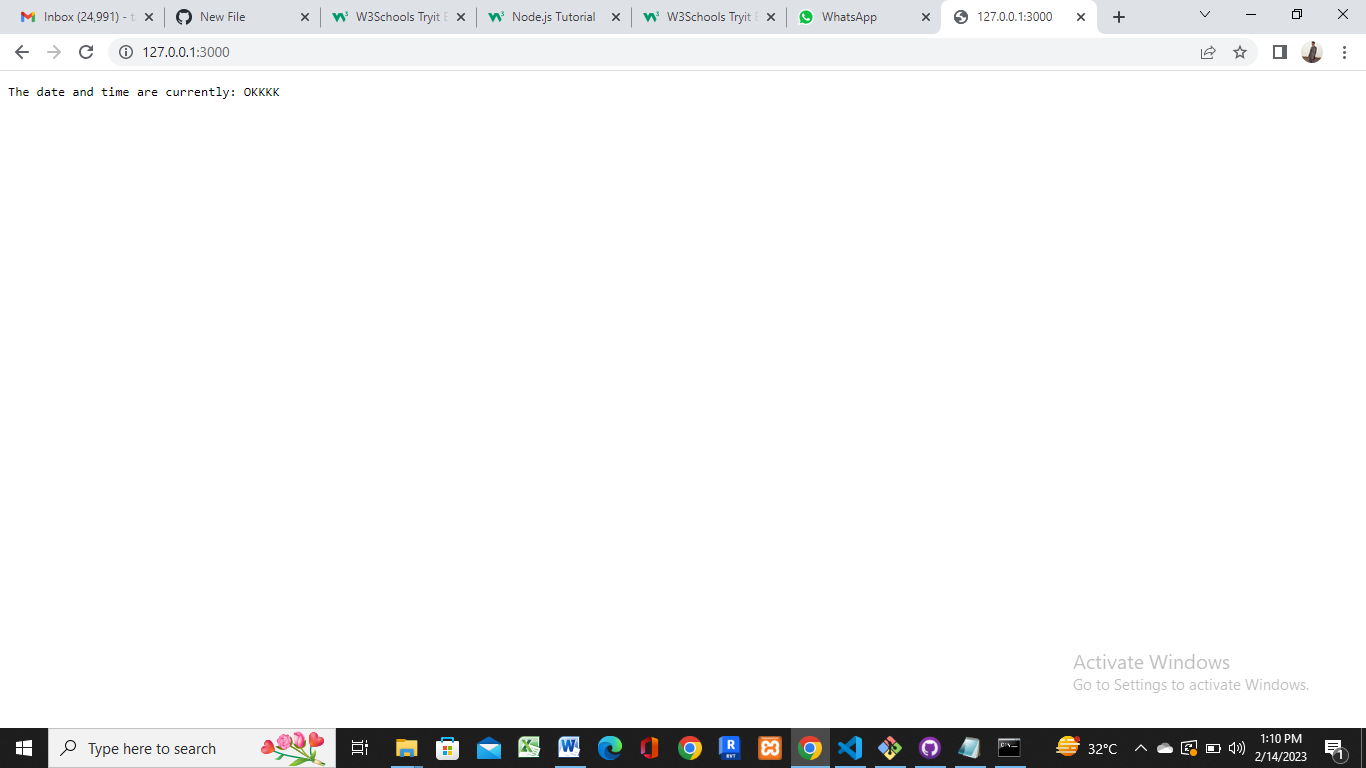
Day:-2

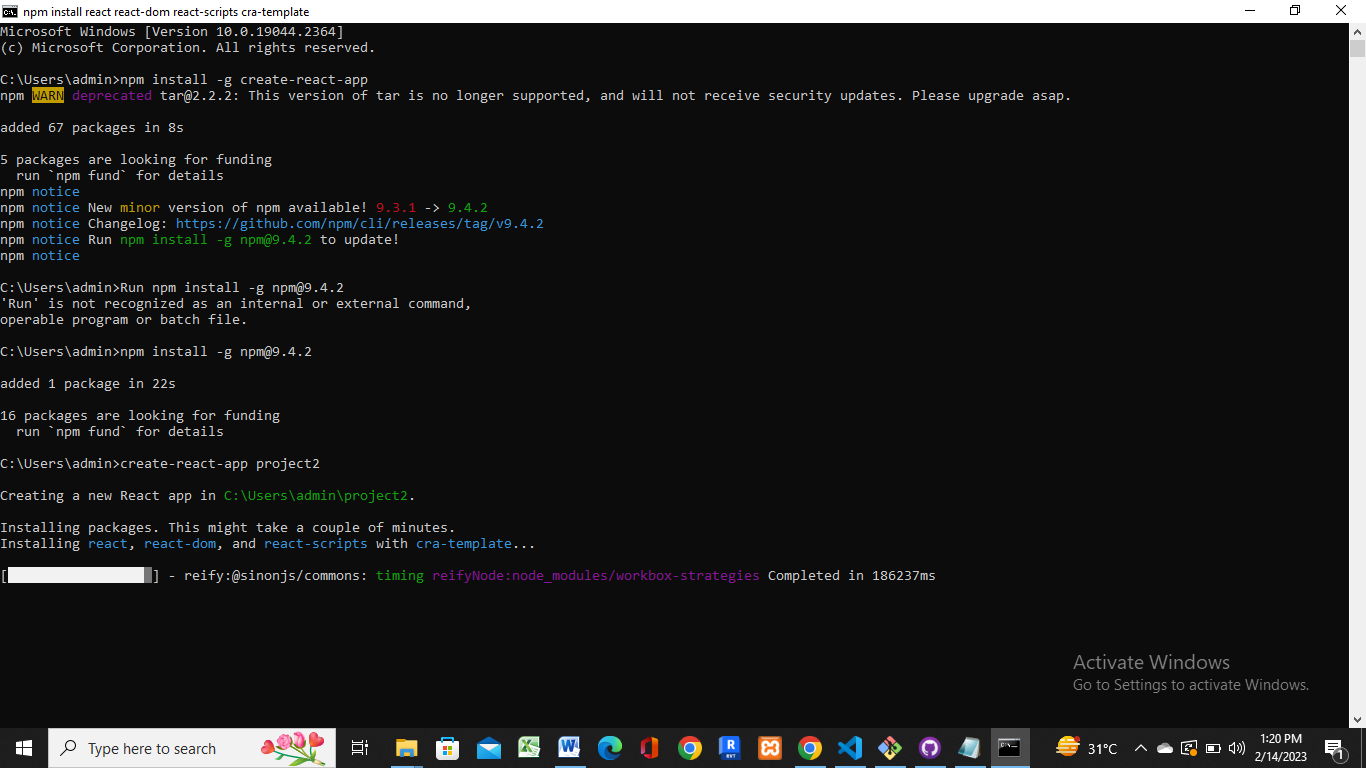


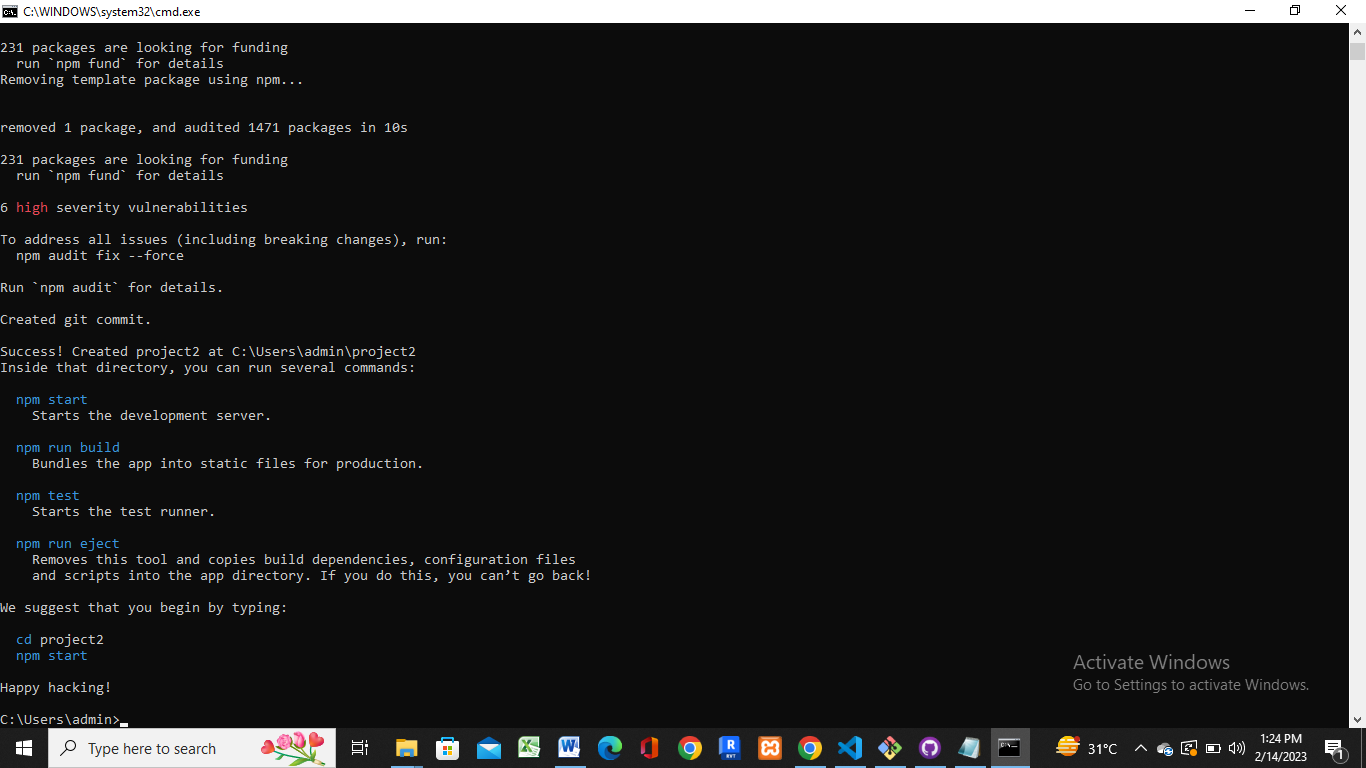


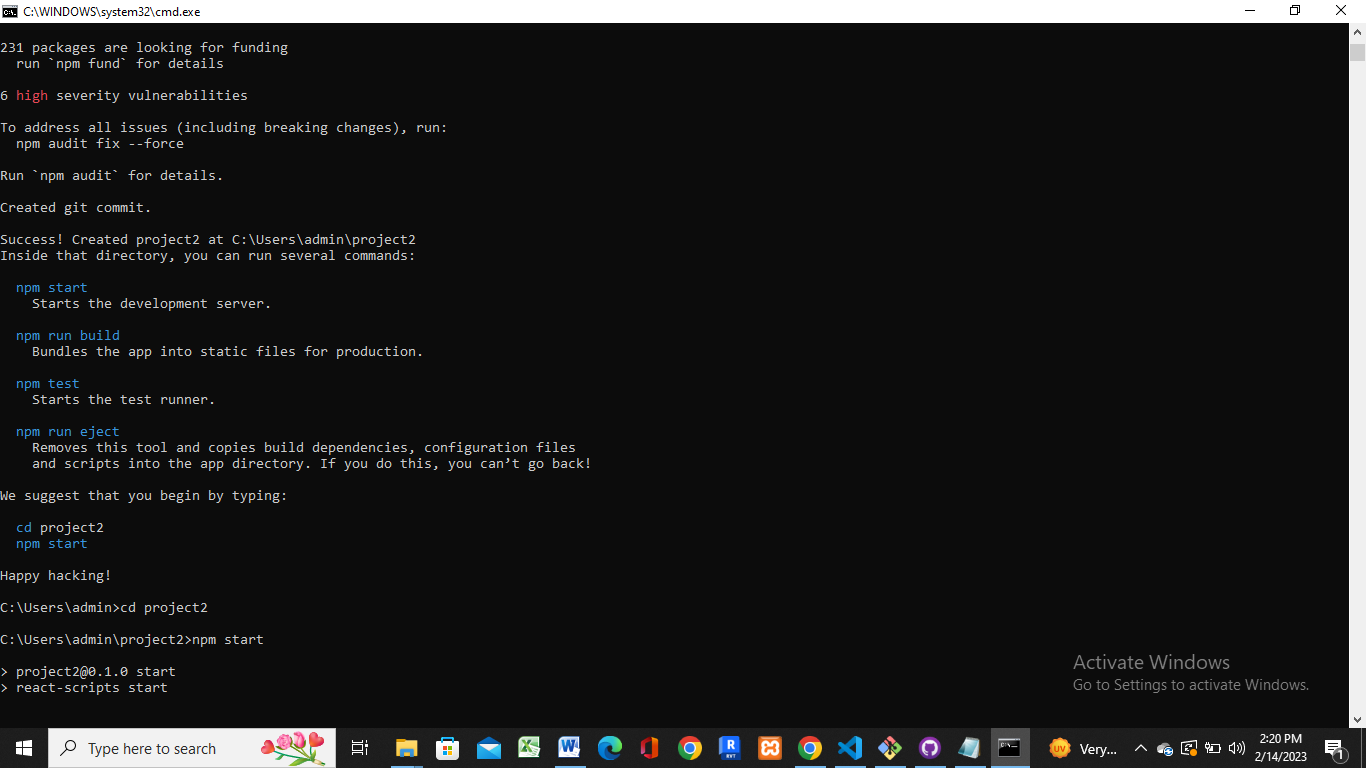
C:\Program Files\nodejs\

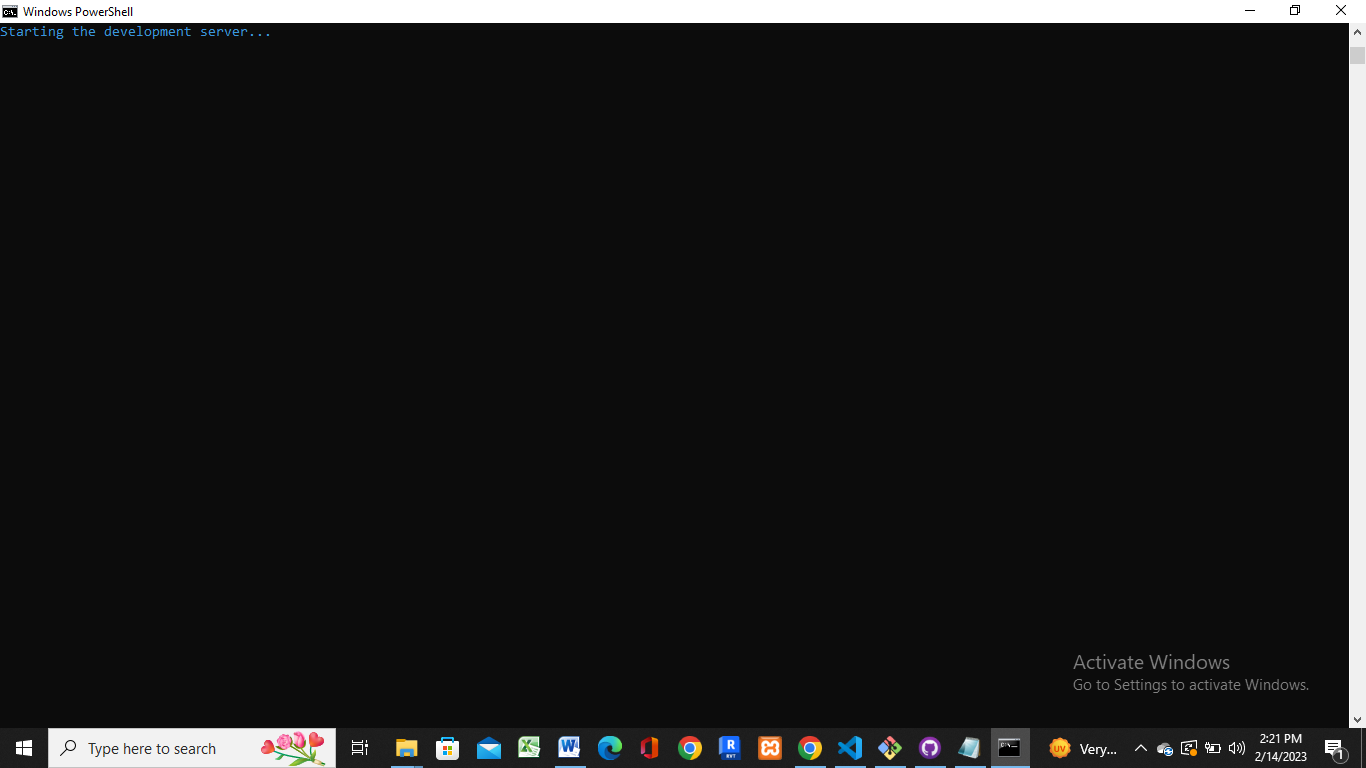


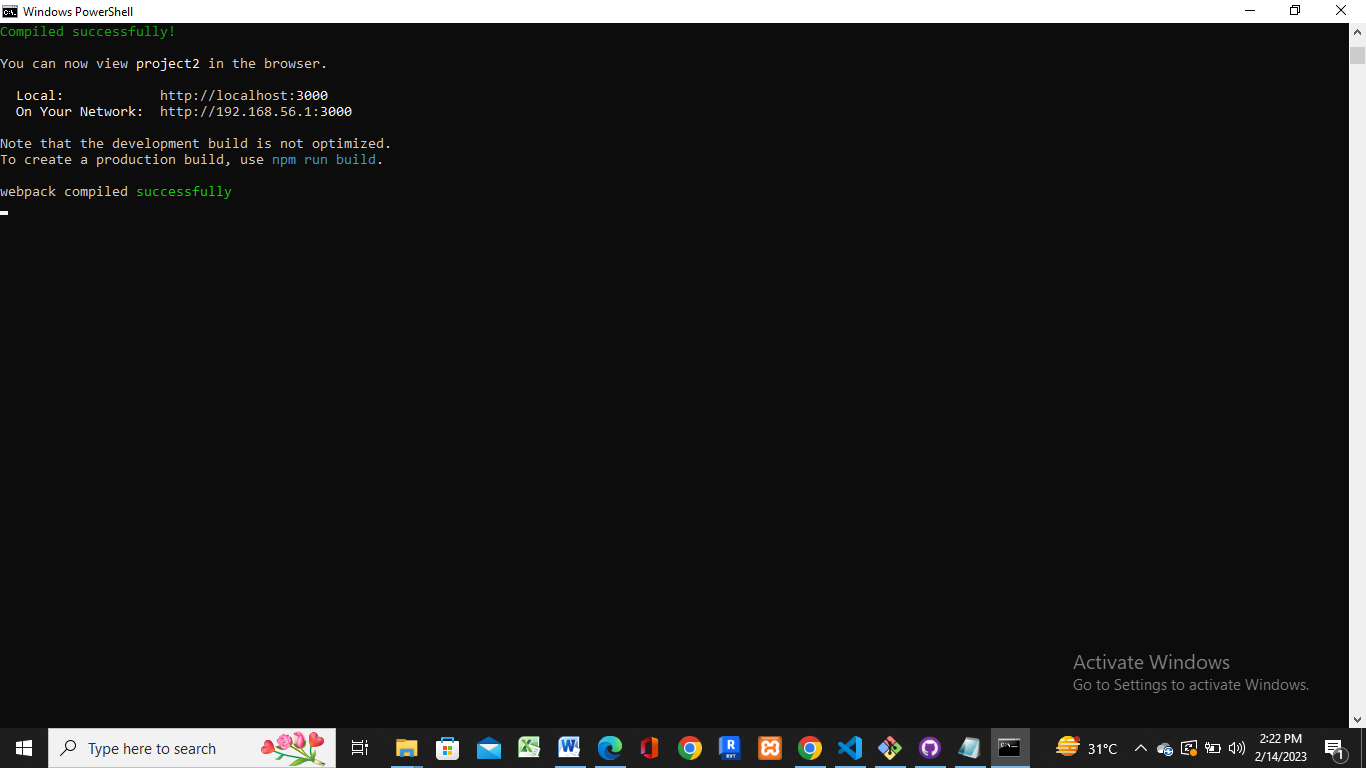
1.Running node js code

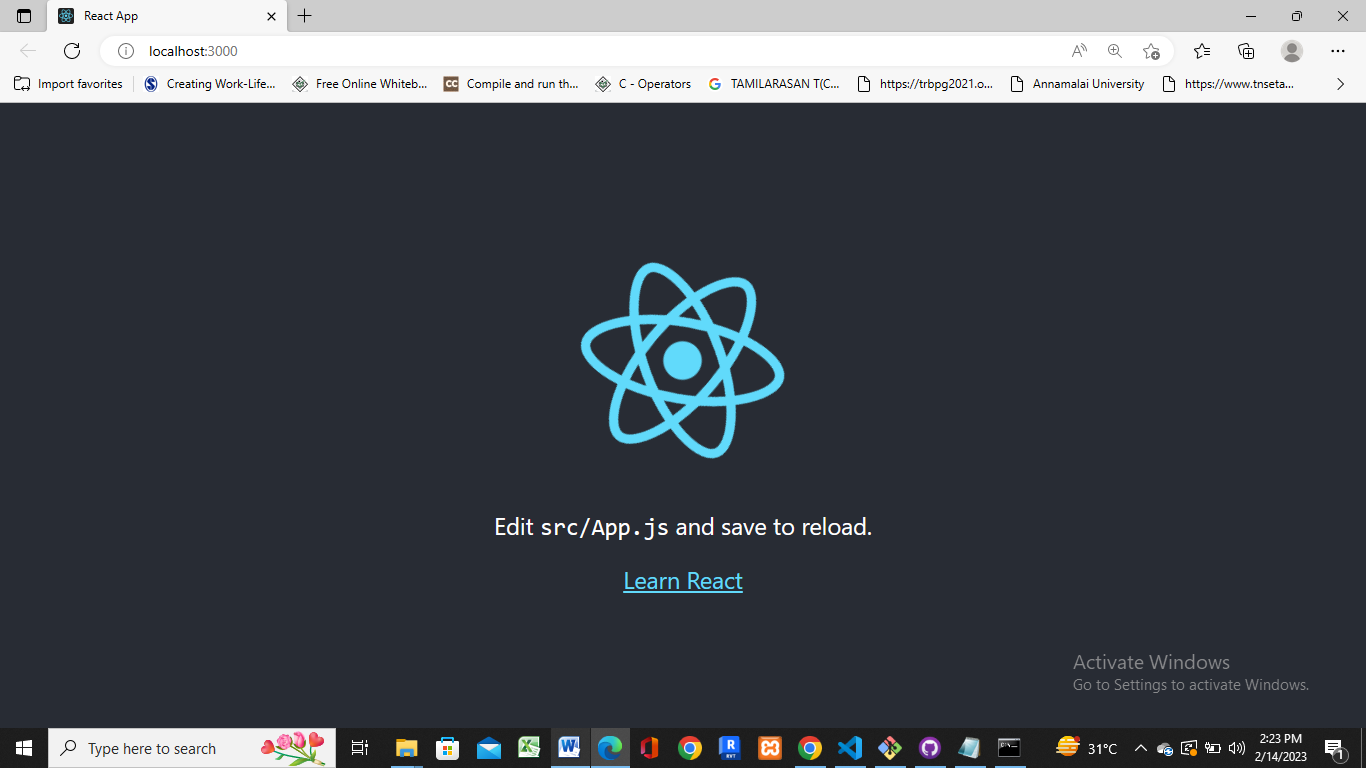


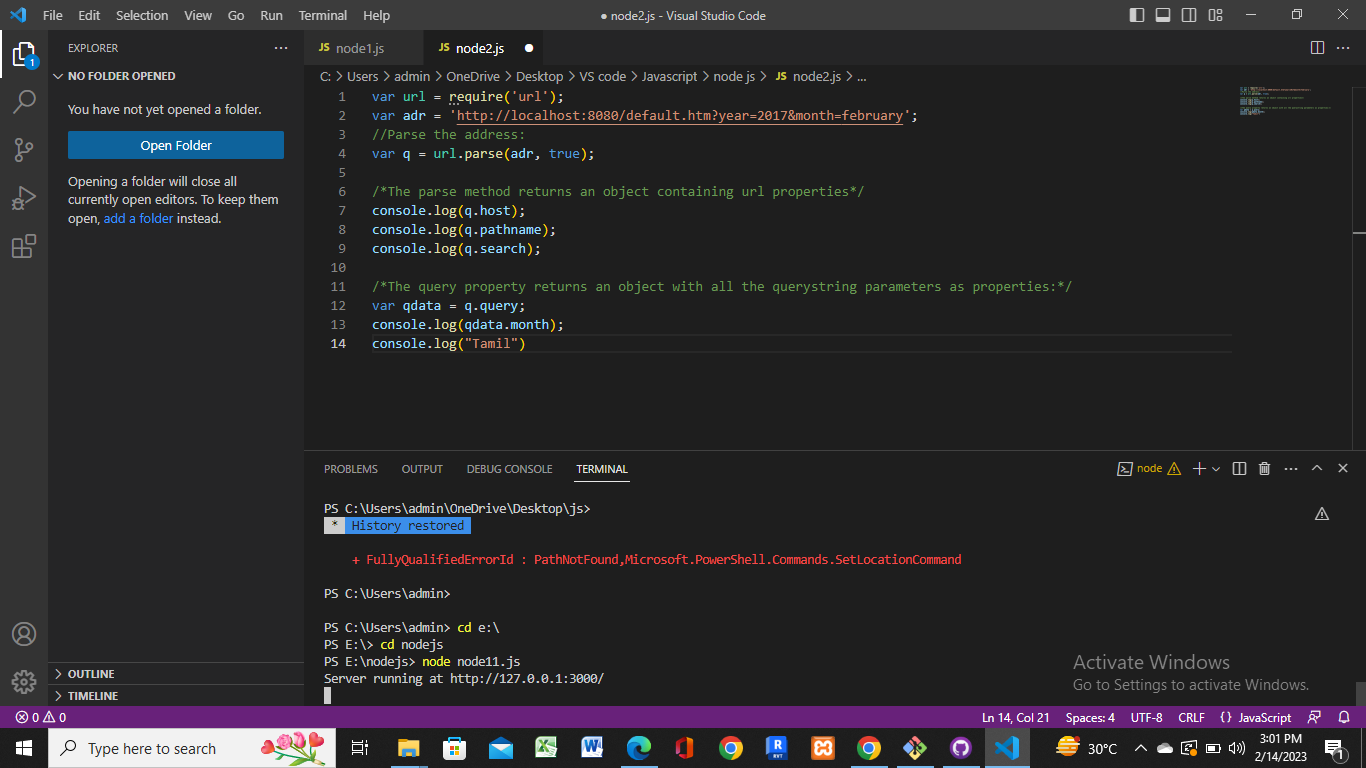


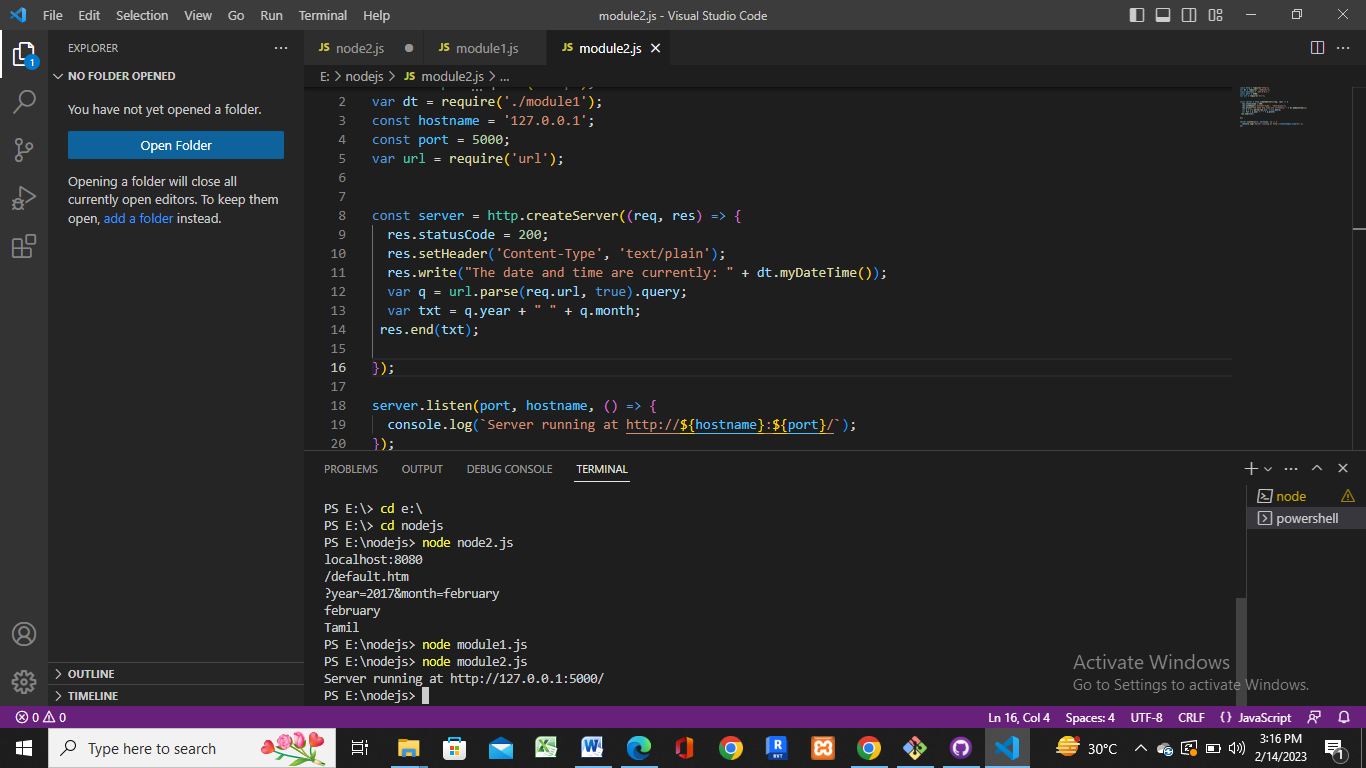


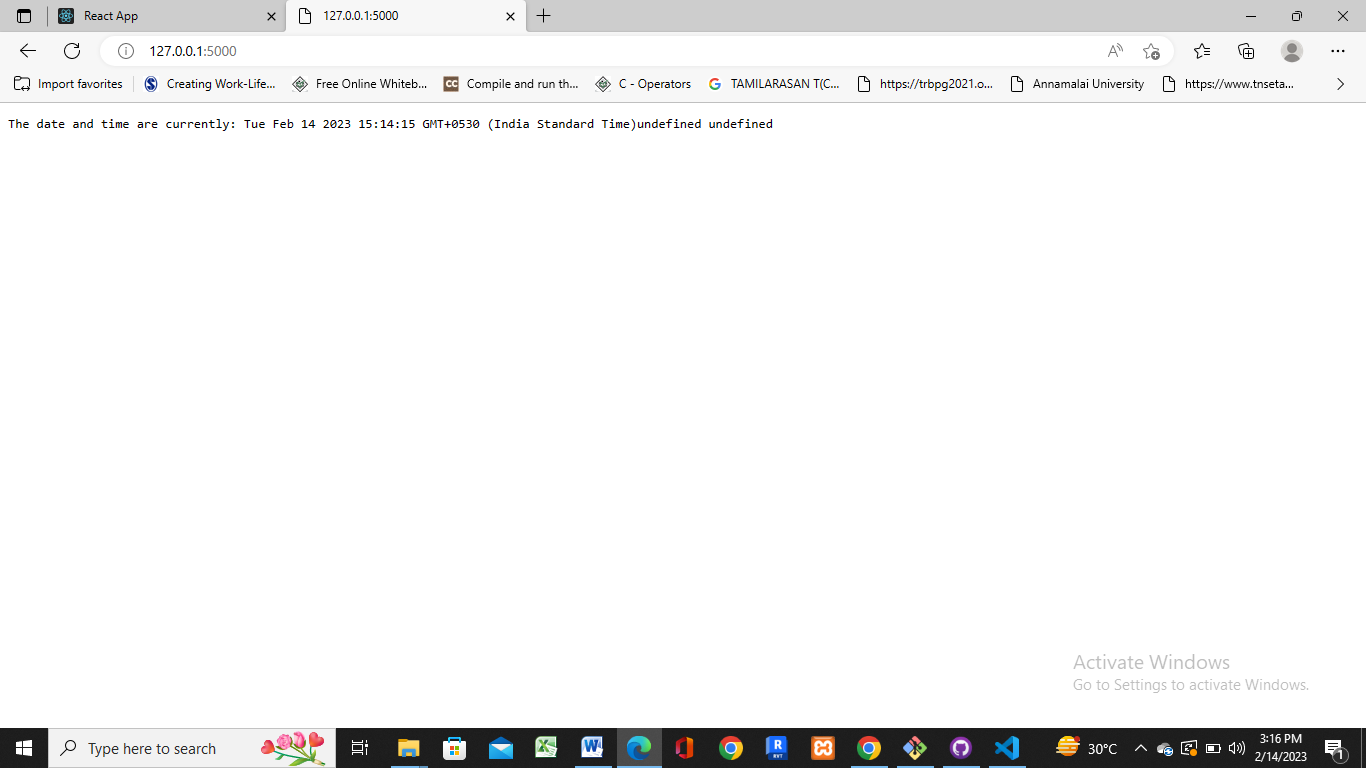


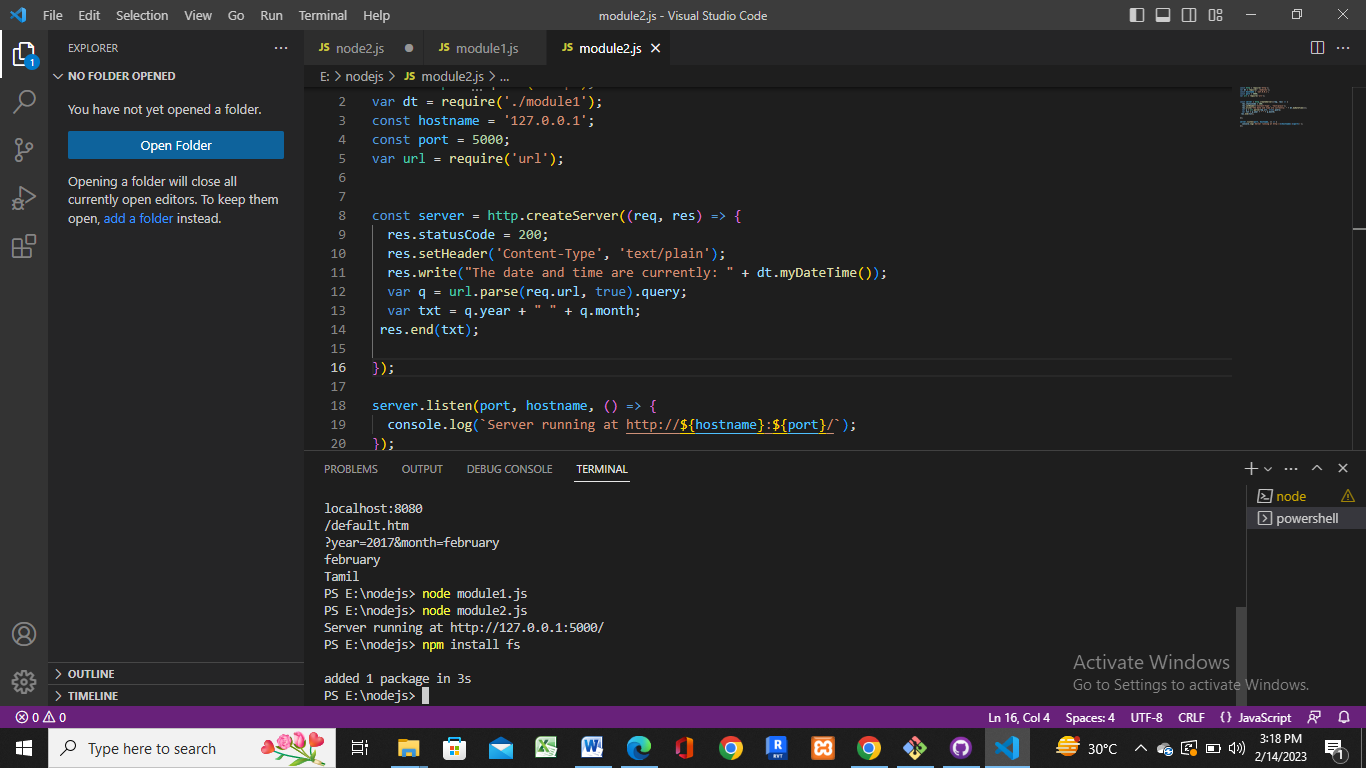


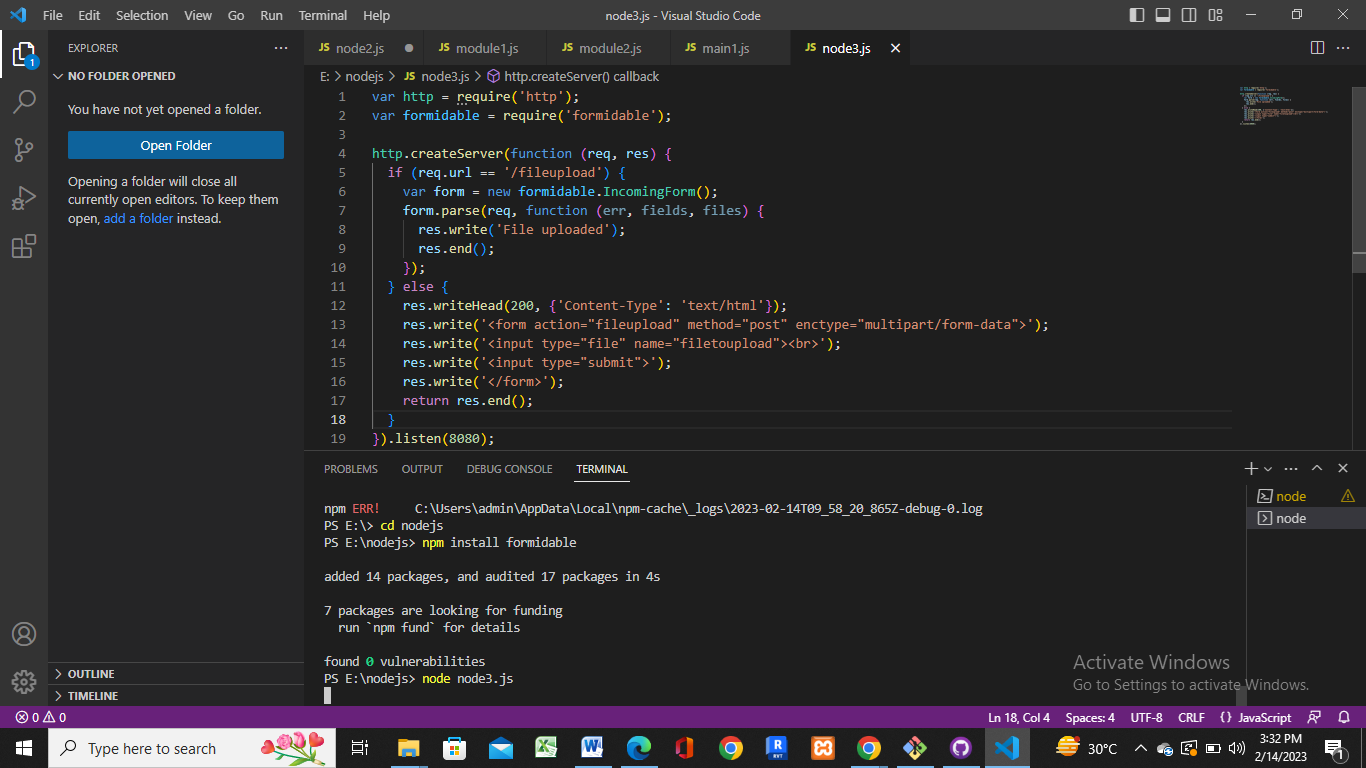


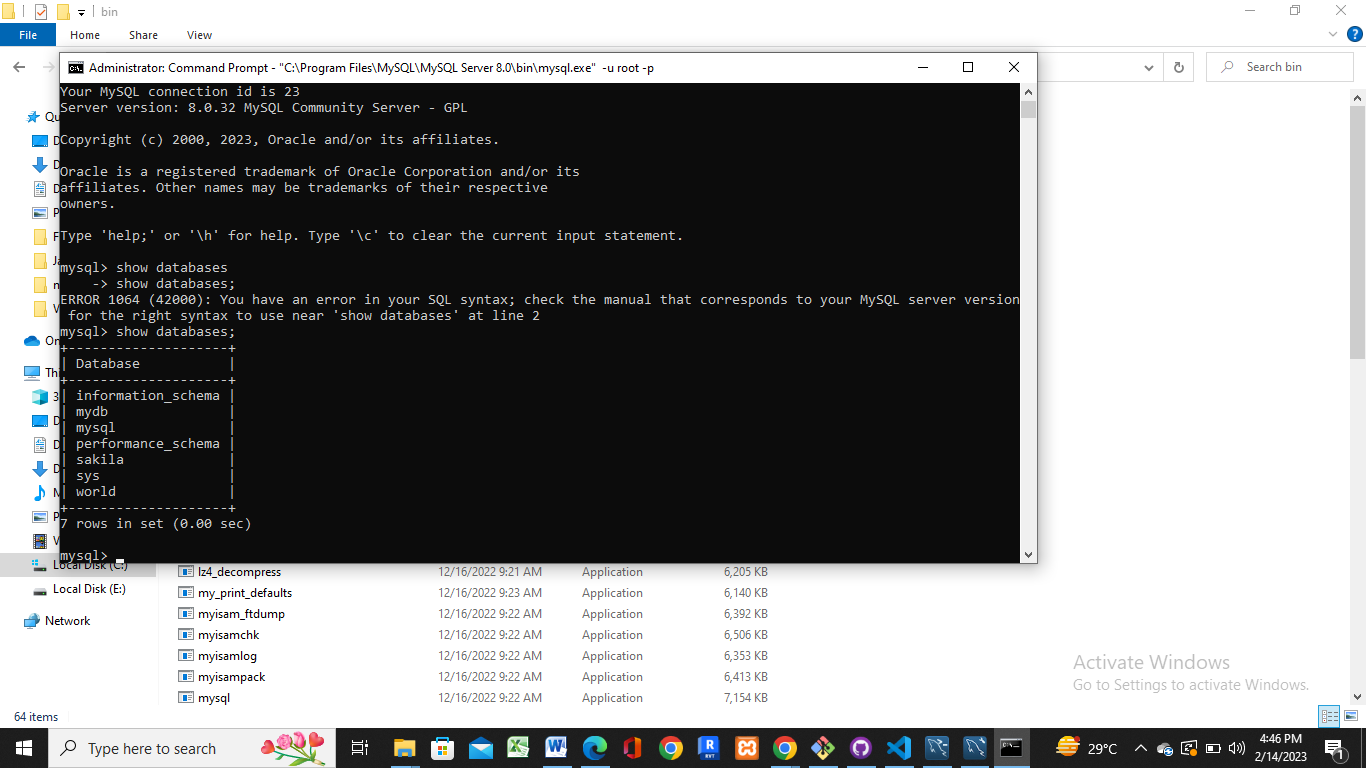


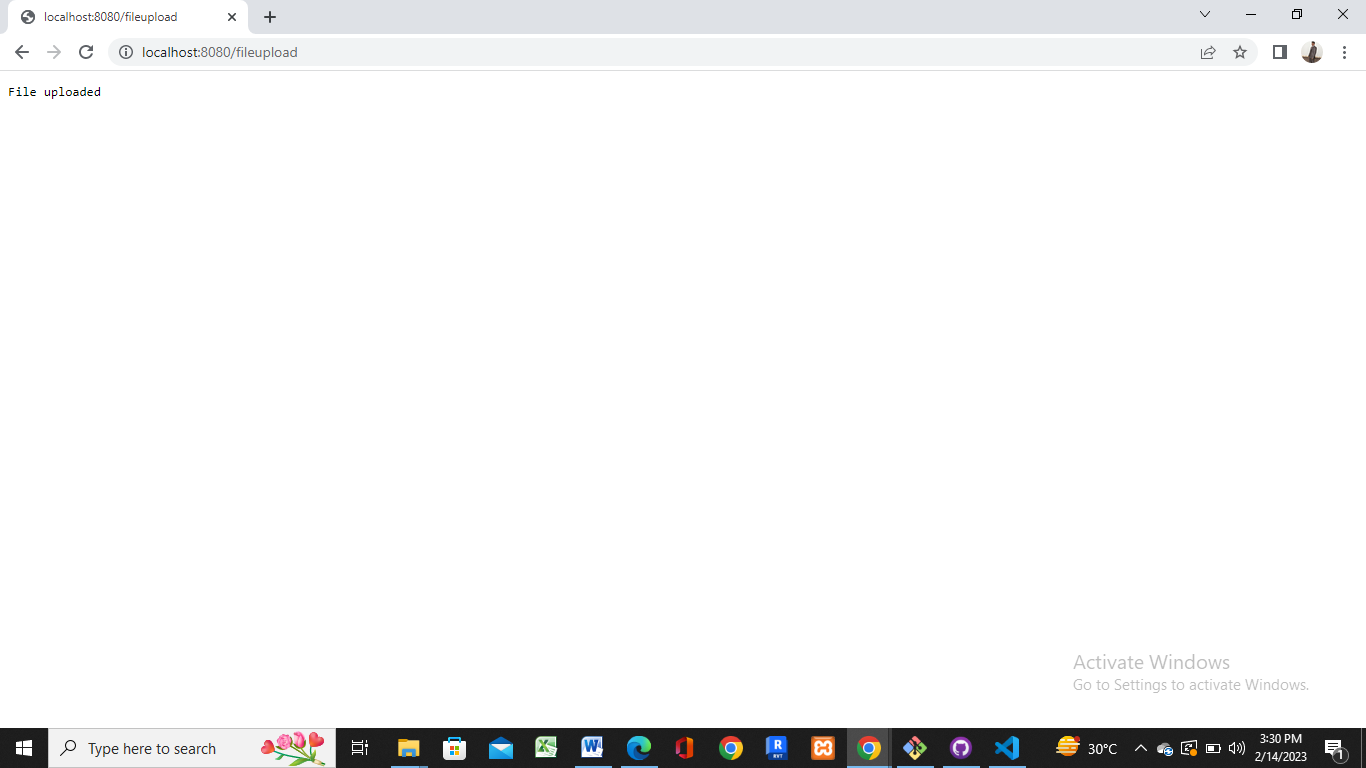




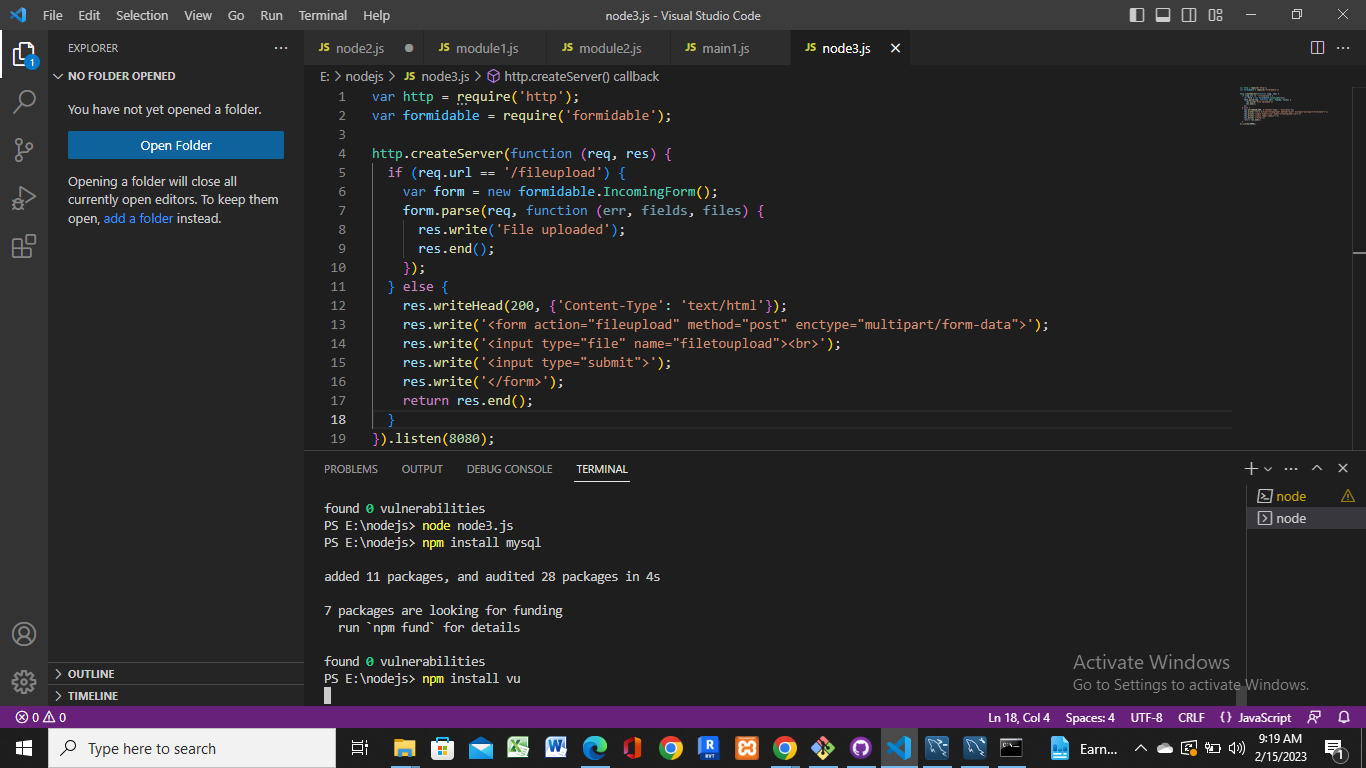








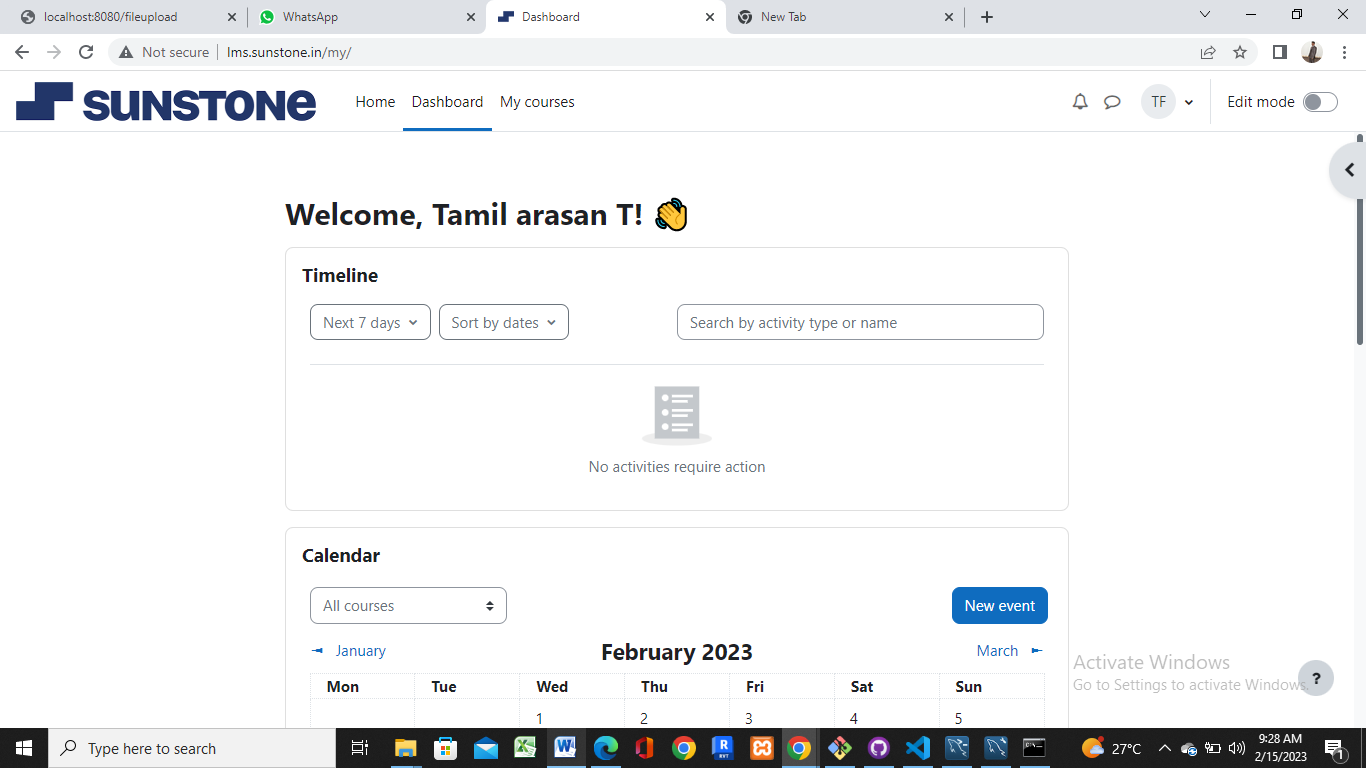
Day 3:-Installing vu



Sunstone login details

User name:panimalar\_tamil

Psw:Tamiluma@87



Topics covered

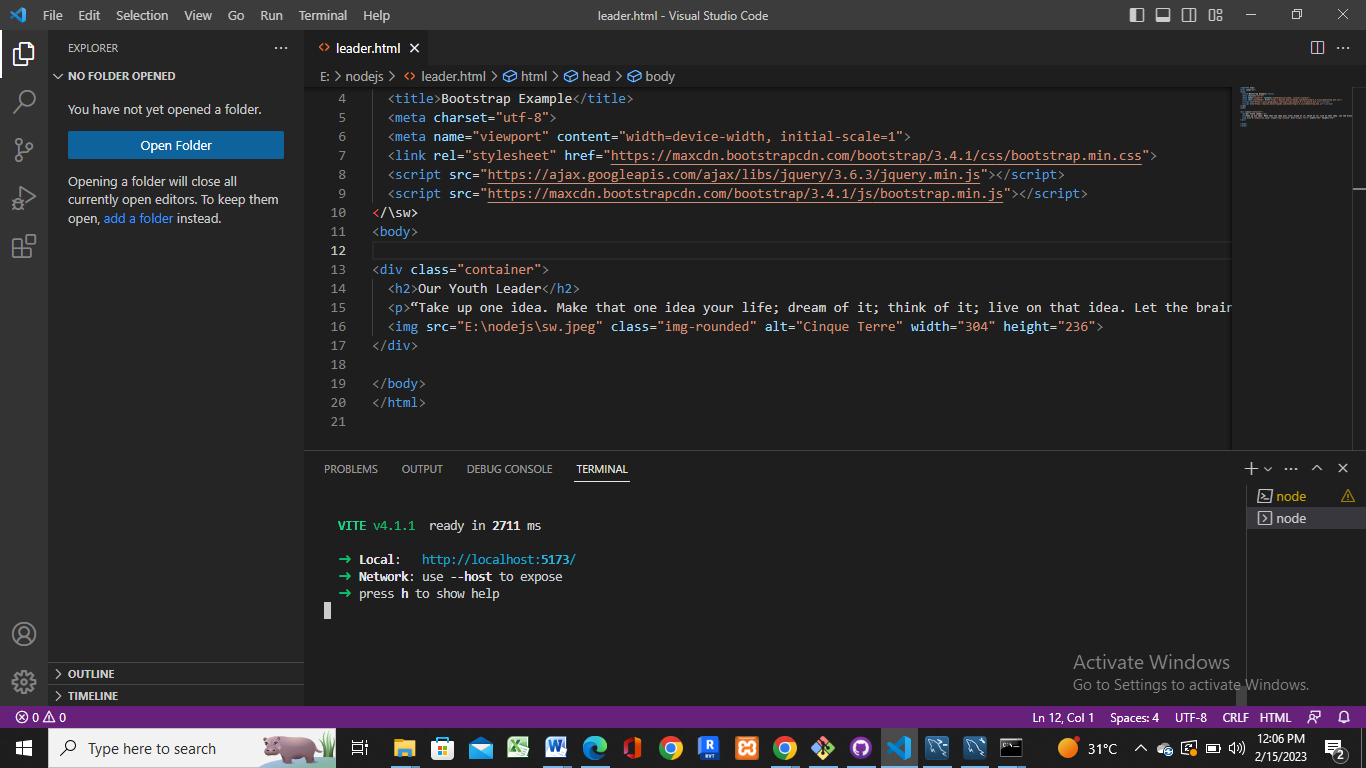
Web client server architecture

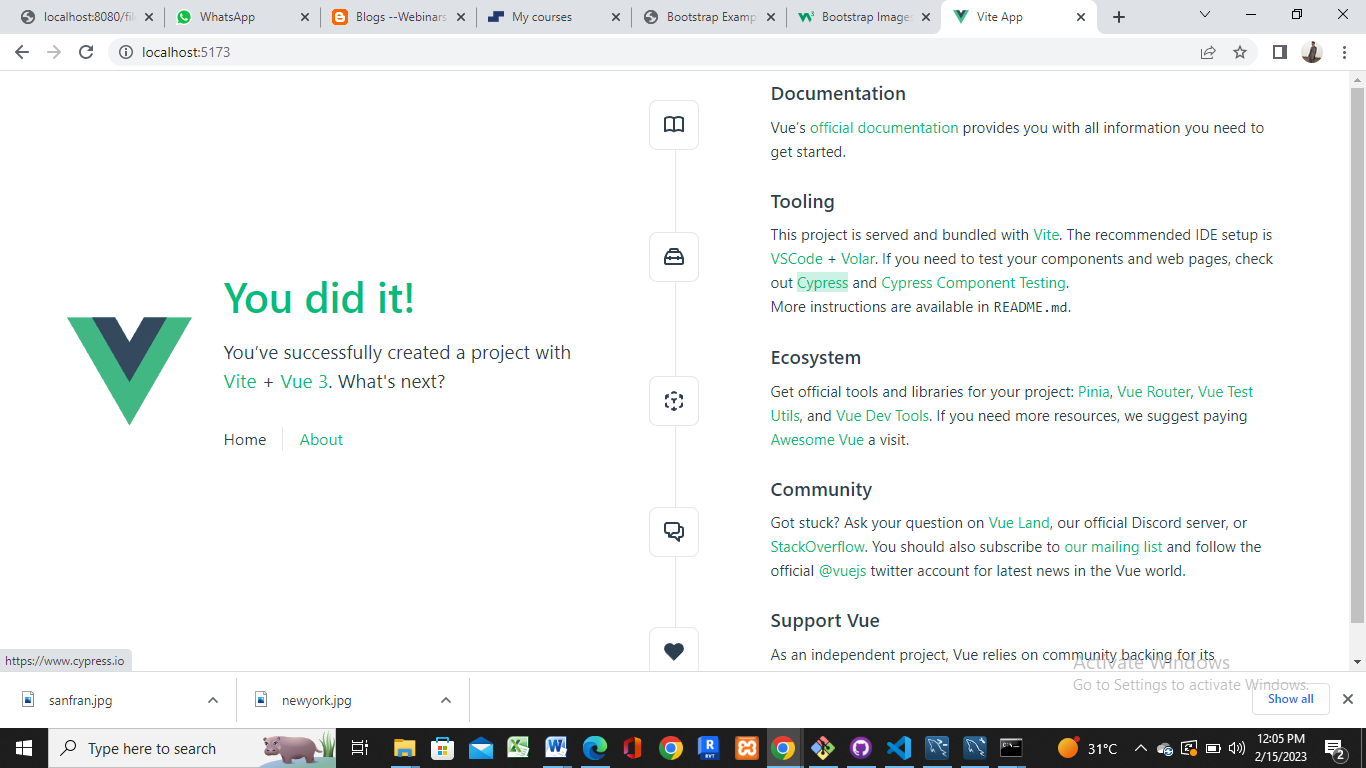
Html structure tags

Bootstrap

Jquery-ajax-promises

Api integration

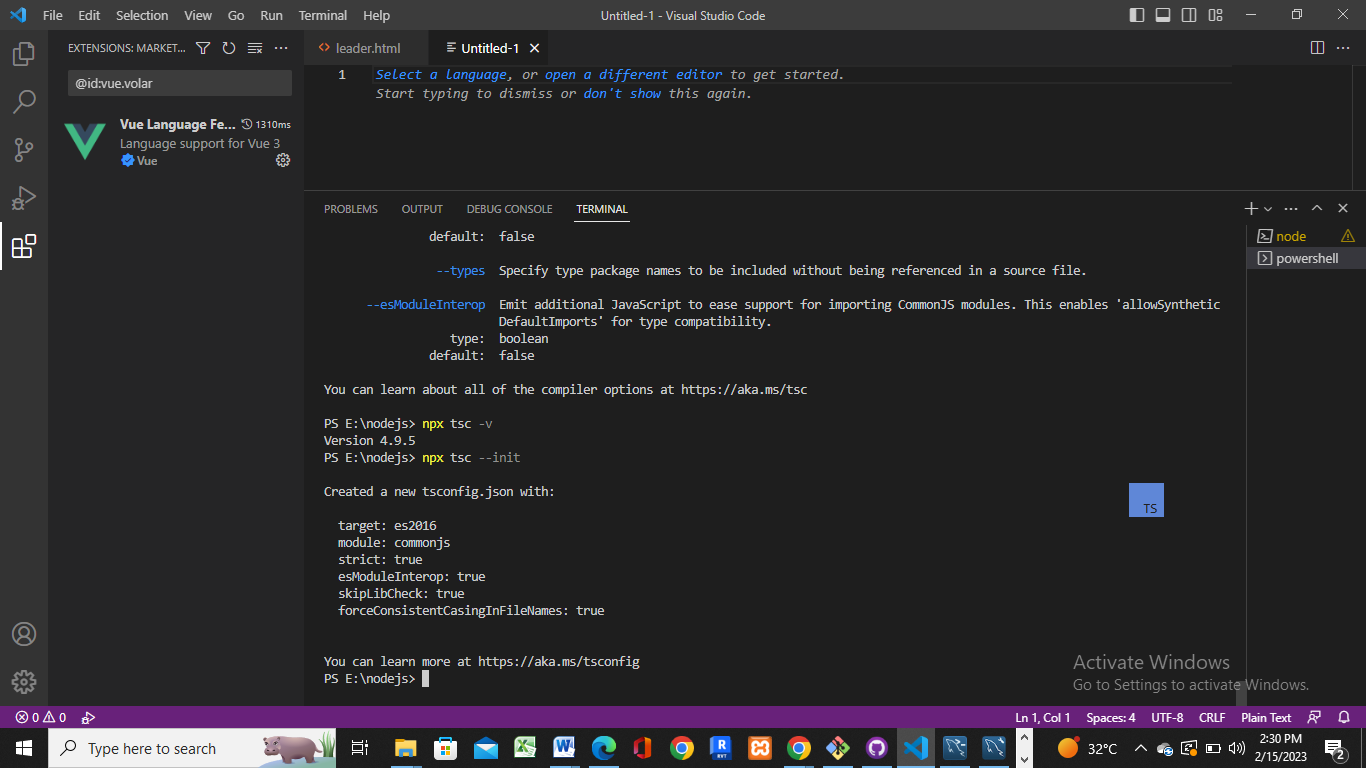


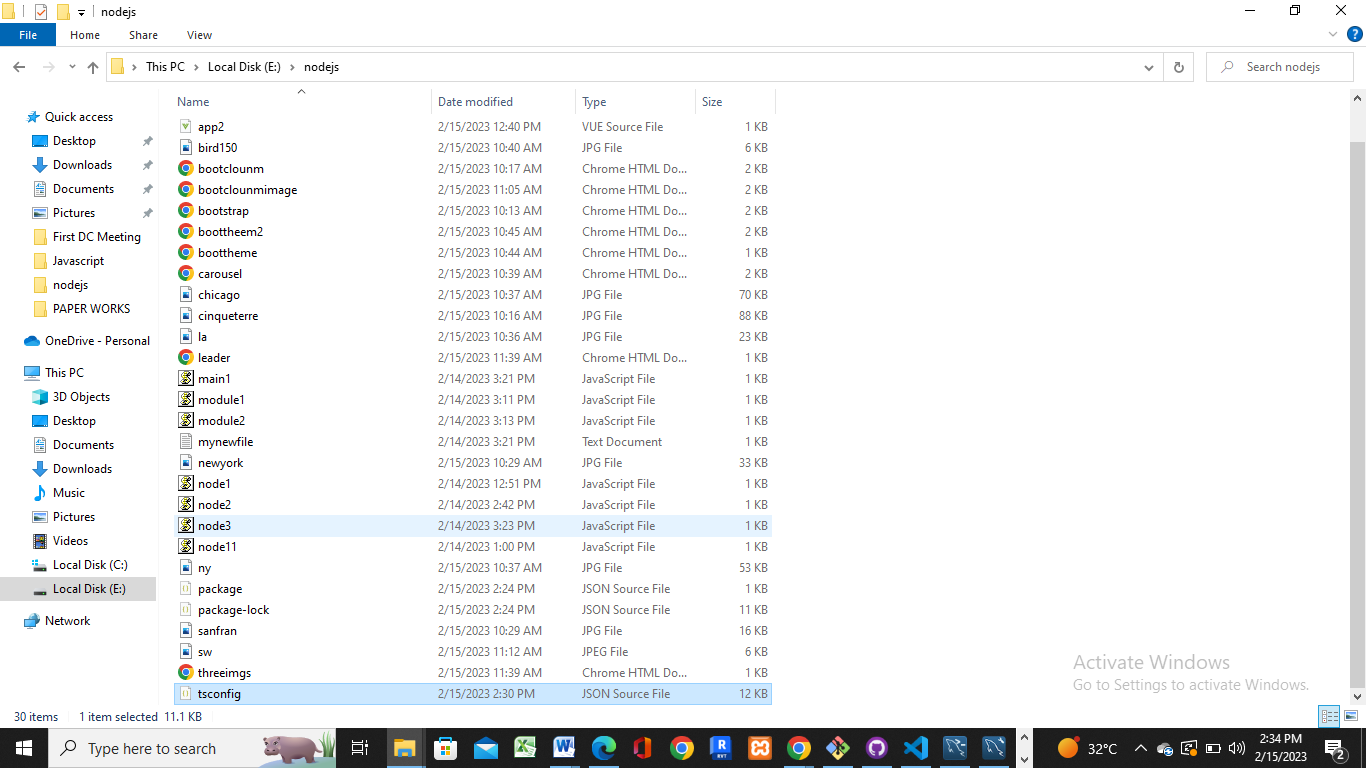


Vue cmds

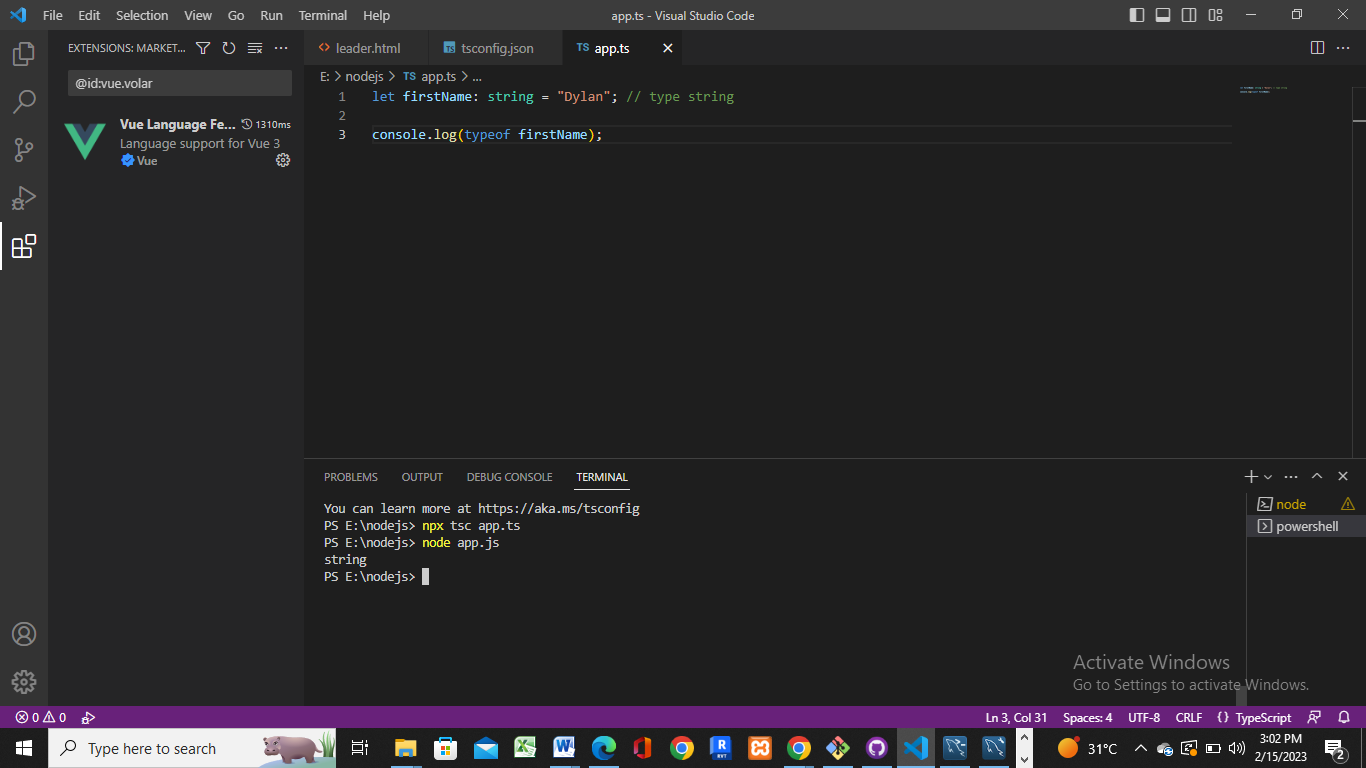
C:\Users\admin\vue-project\src

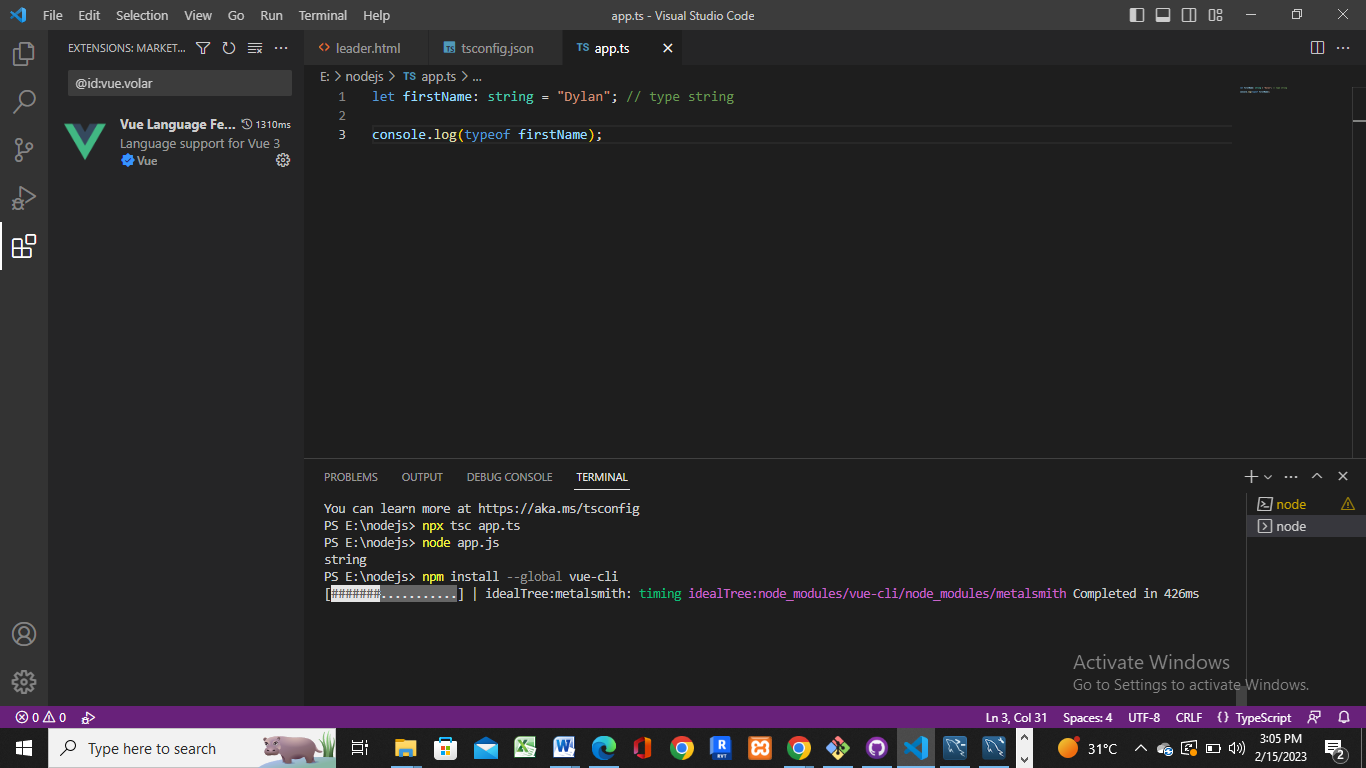
C:\Tamil\router-page\src

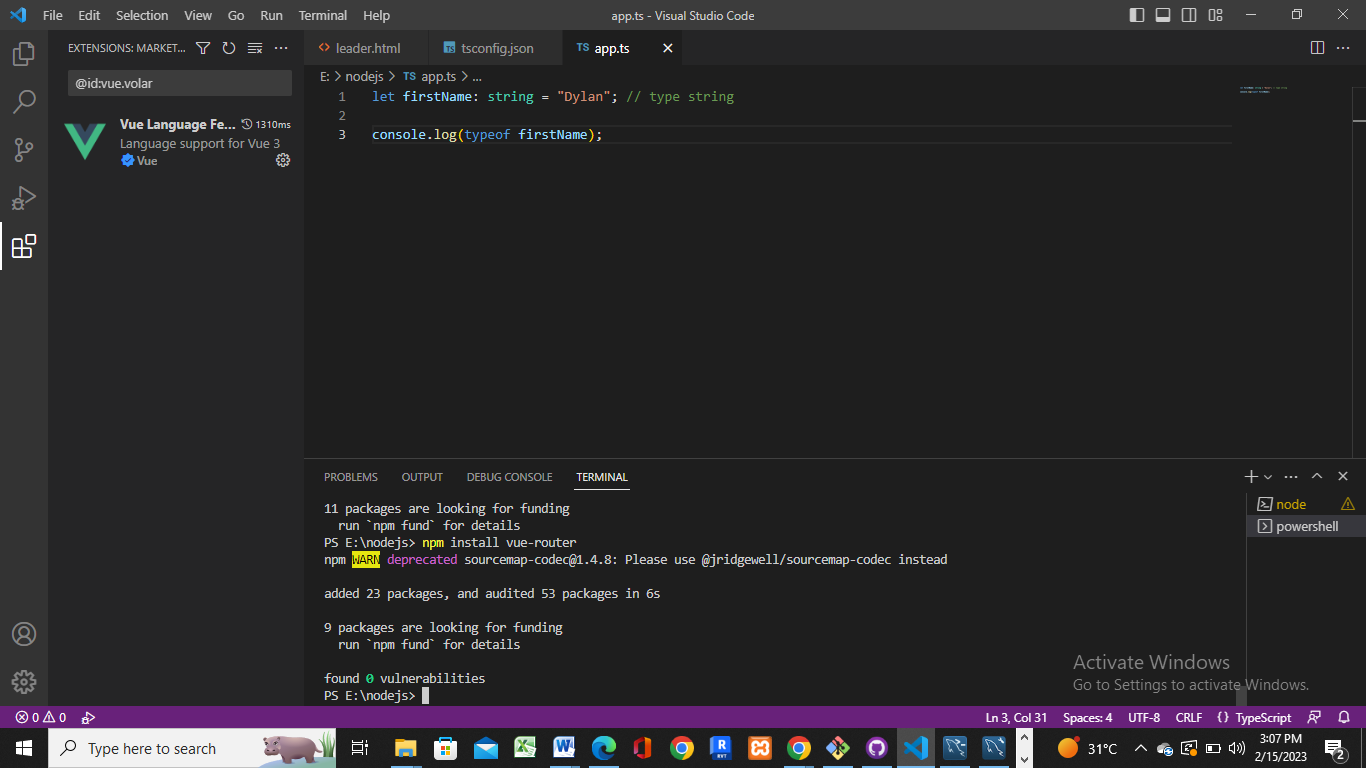


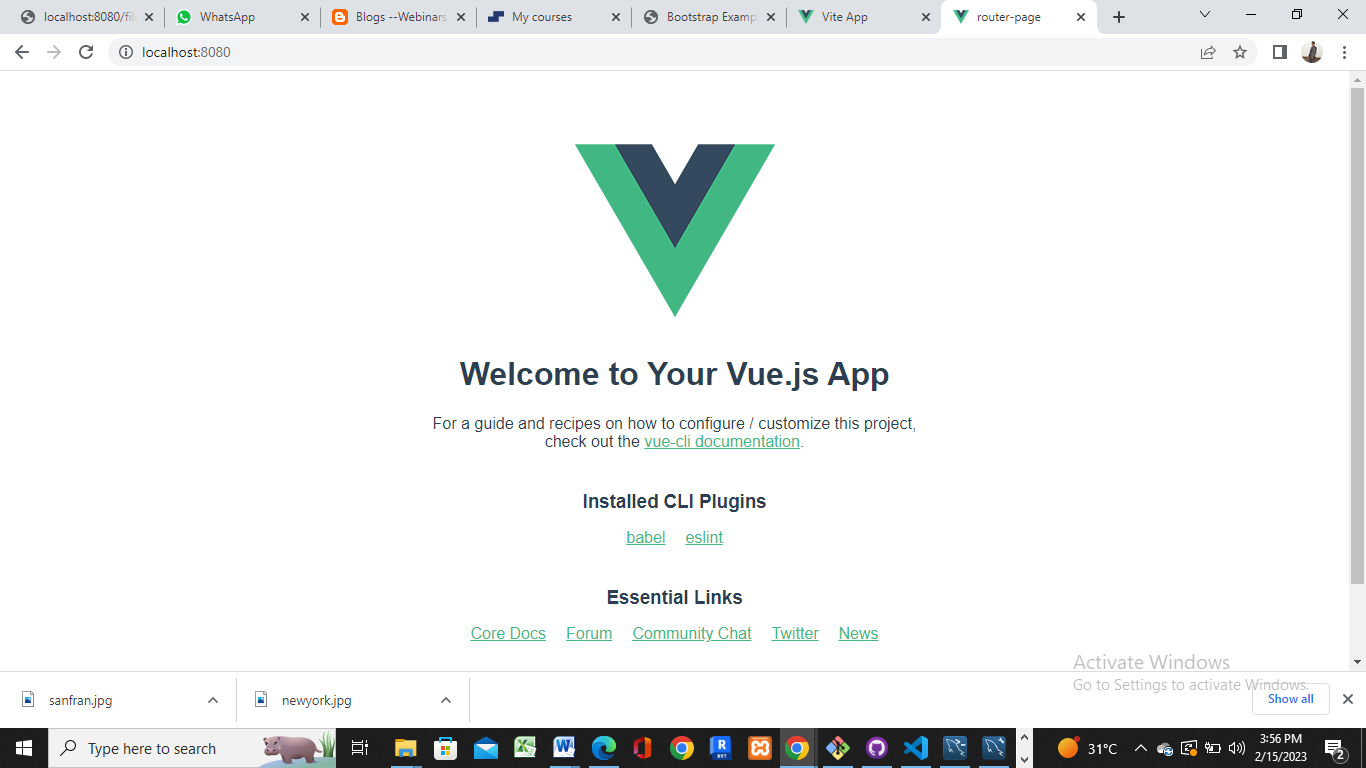


Ts compile and run





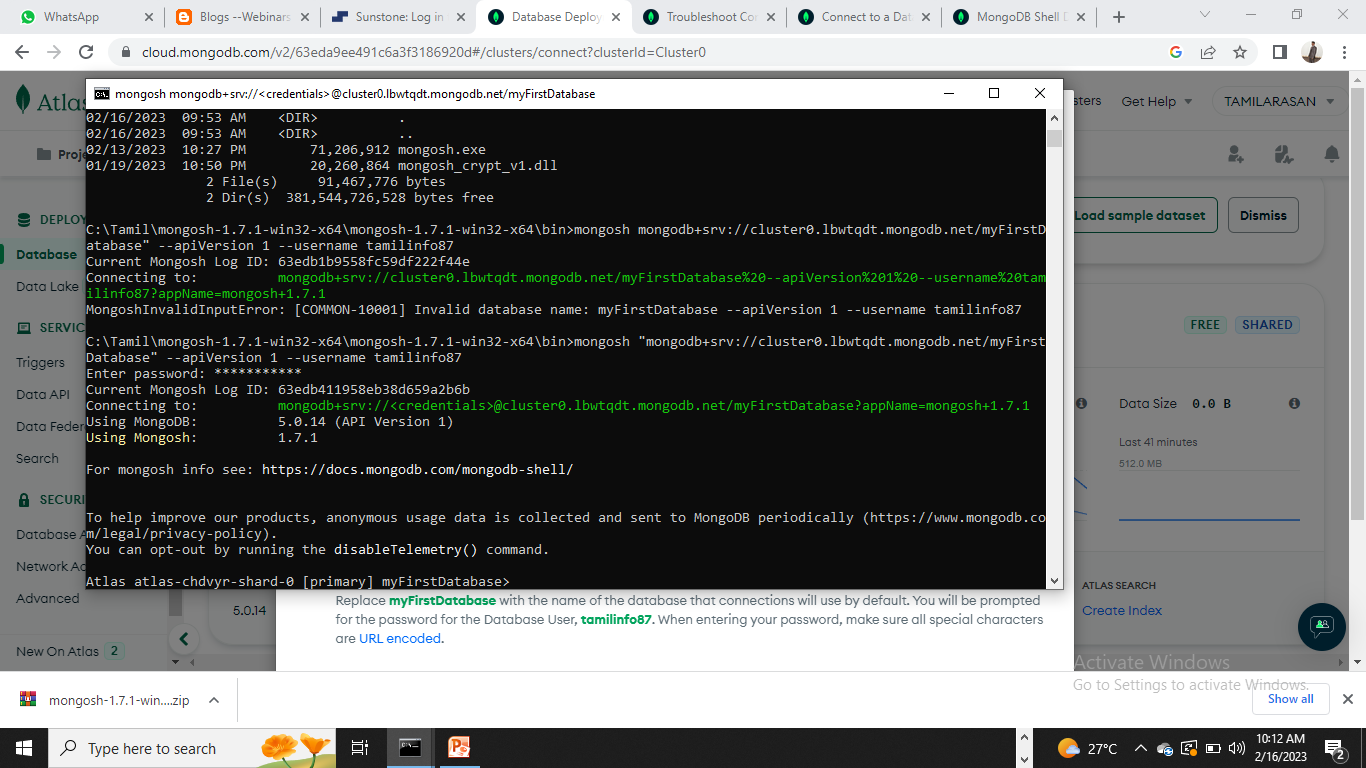


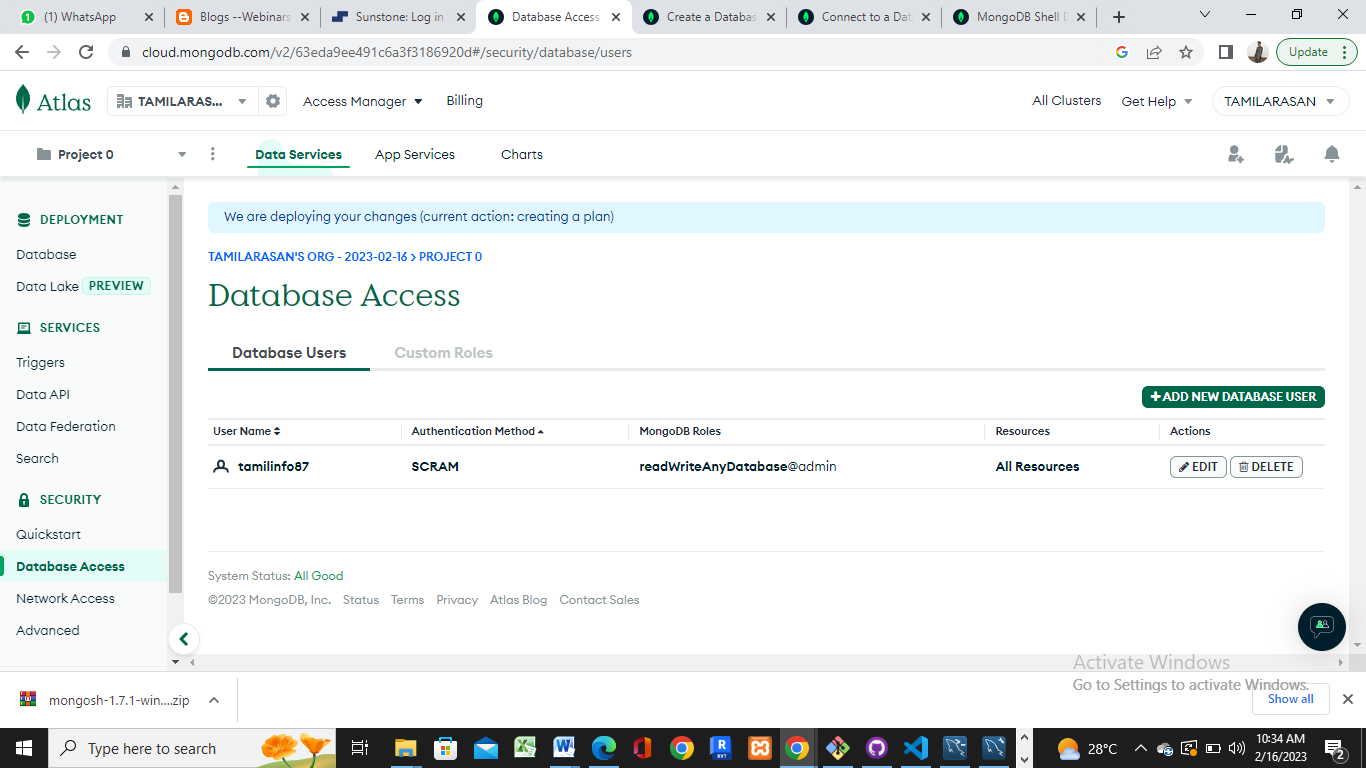


Day 4 mango DB

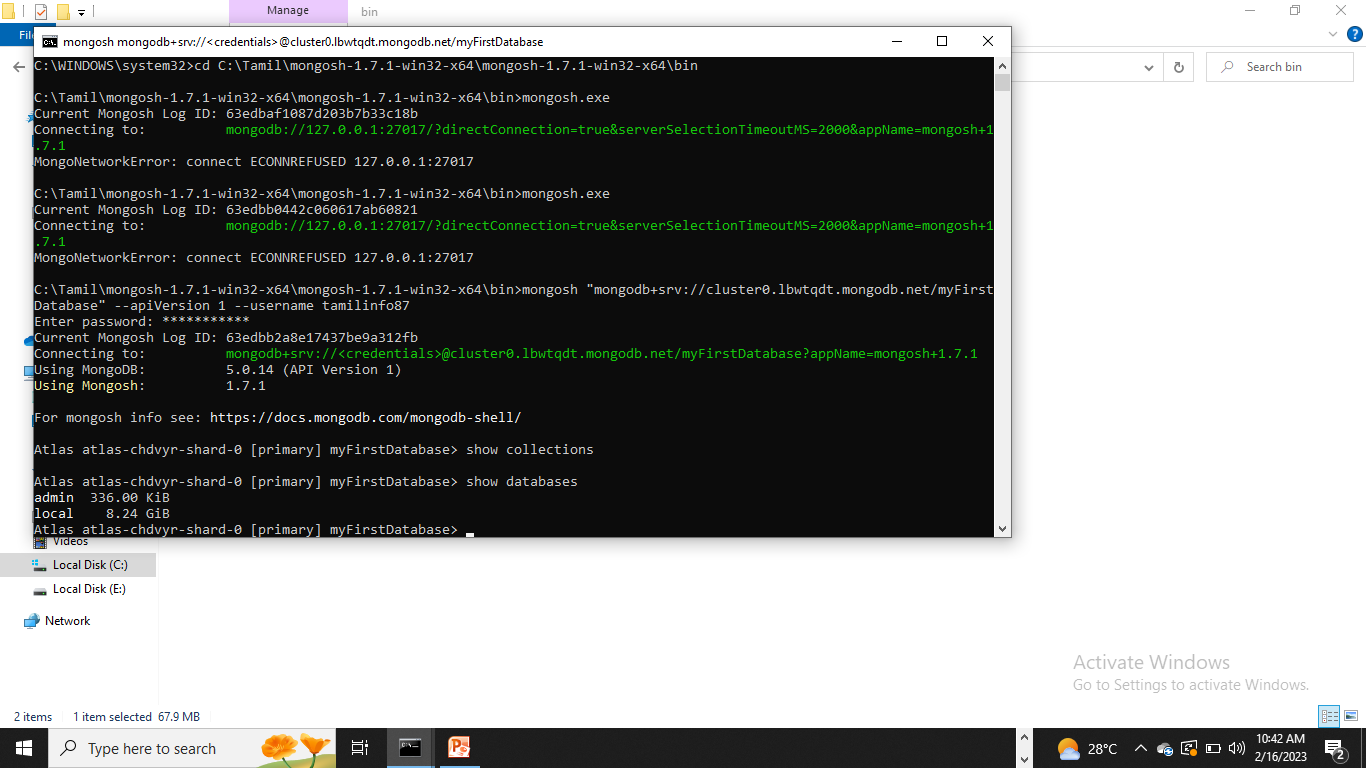
<https://ap-south-1.aws.data.mongodb-api.com/app/data-waxtc/endpoint/data/v1>

mongodb+srv://cluster0.lbwtqdt.mongodb.net/myFirstDatabase" --apiVersion 1 --username tamilinfo87

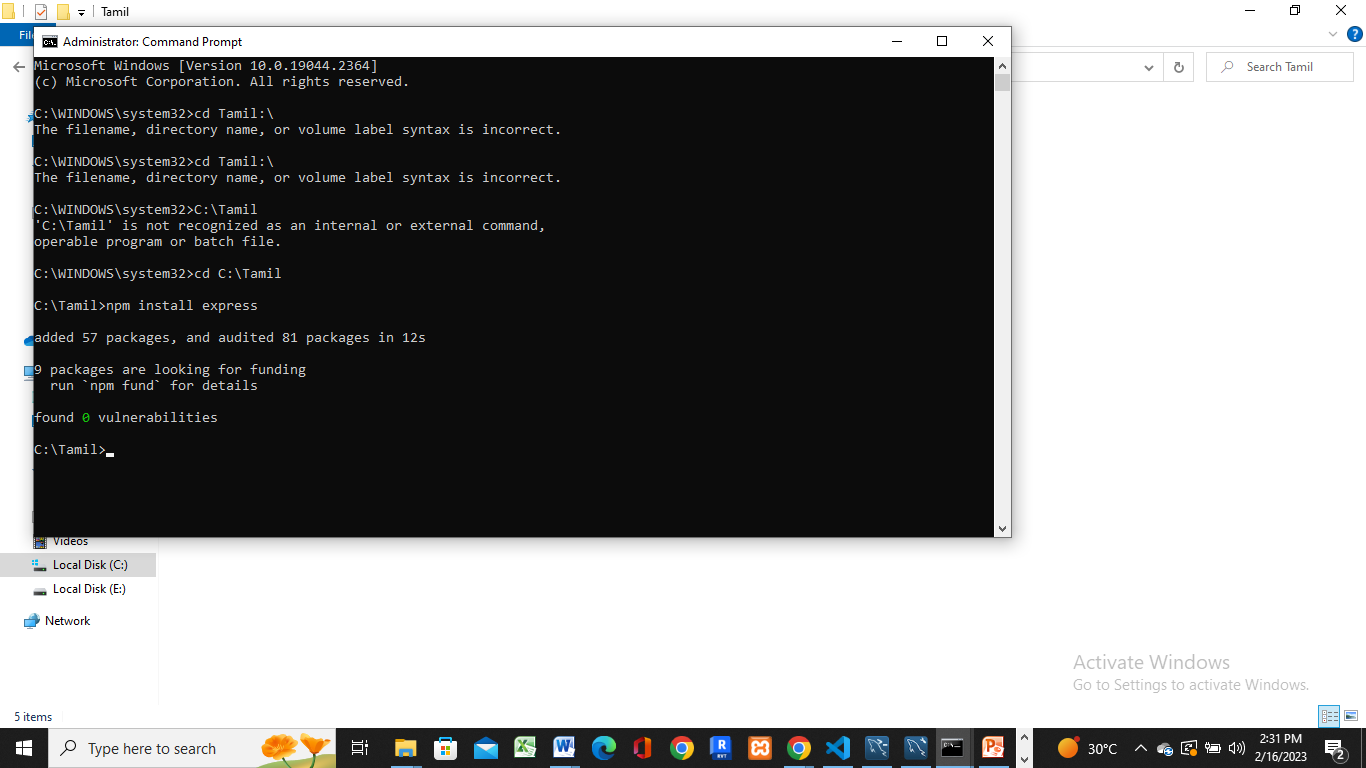


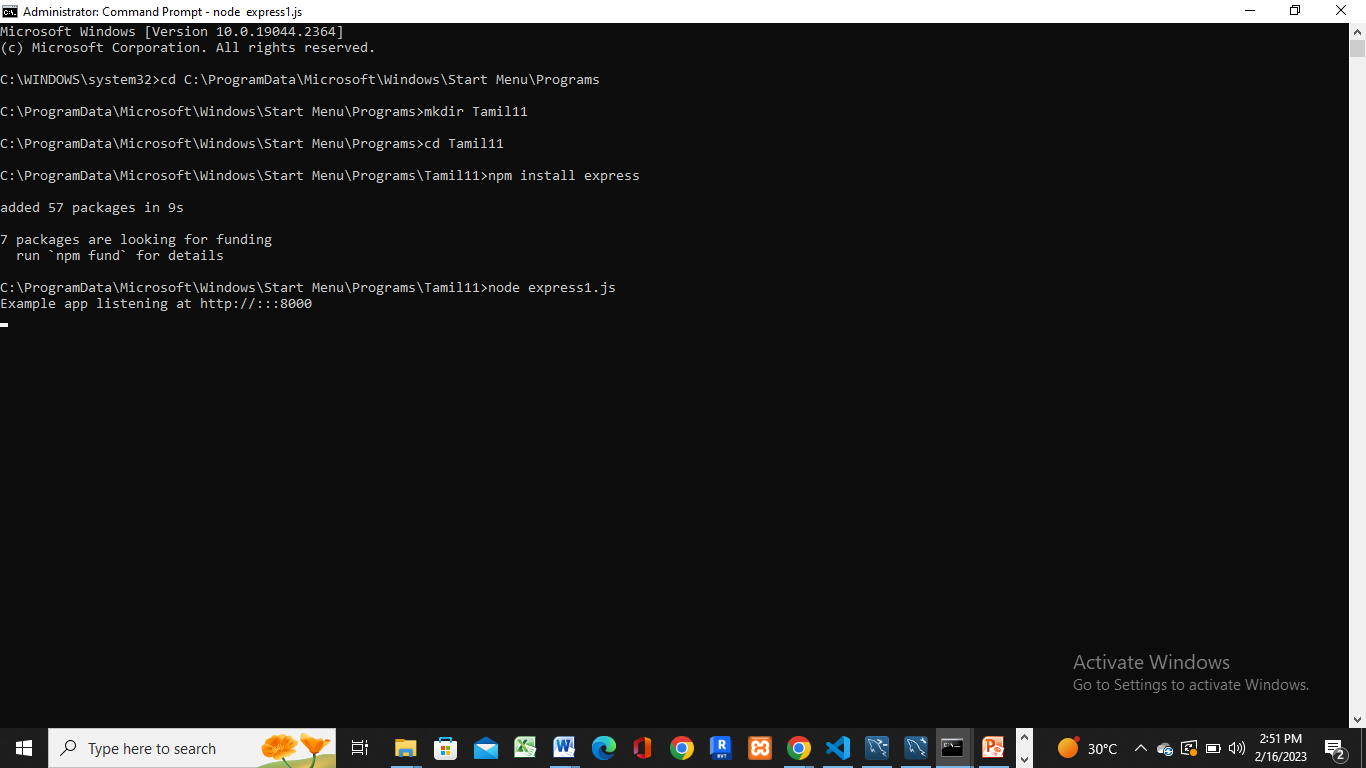


Reconnectiong databases

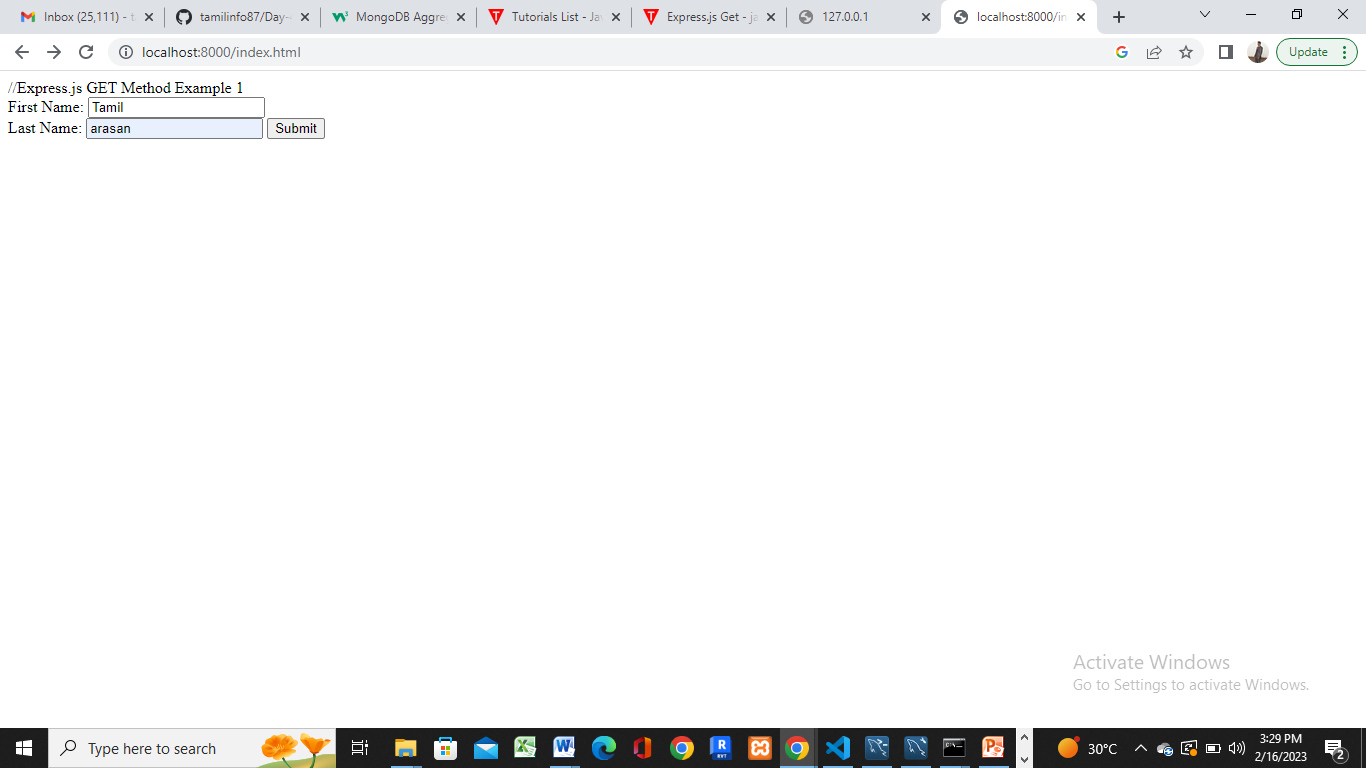


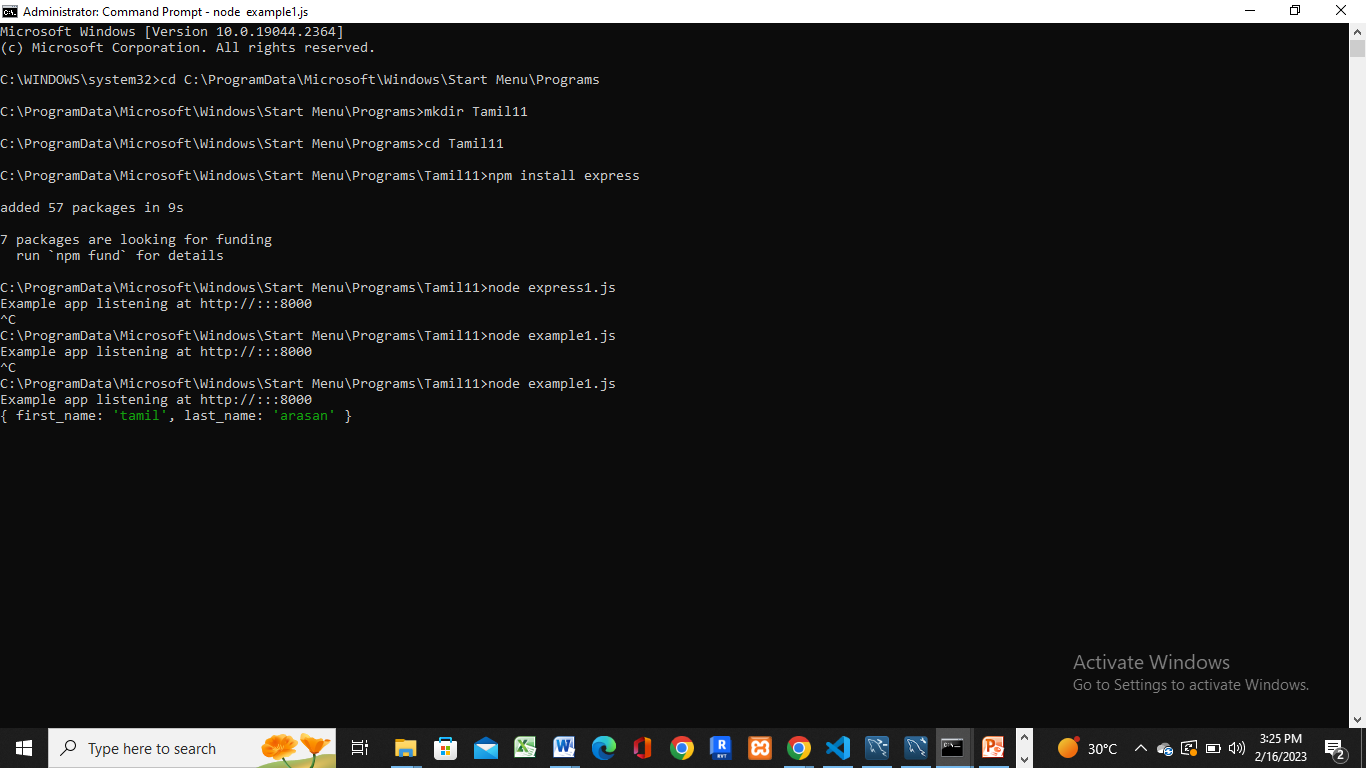
Installing express

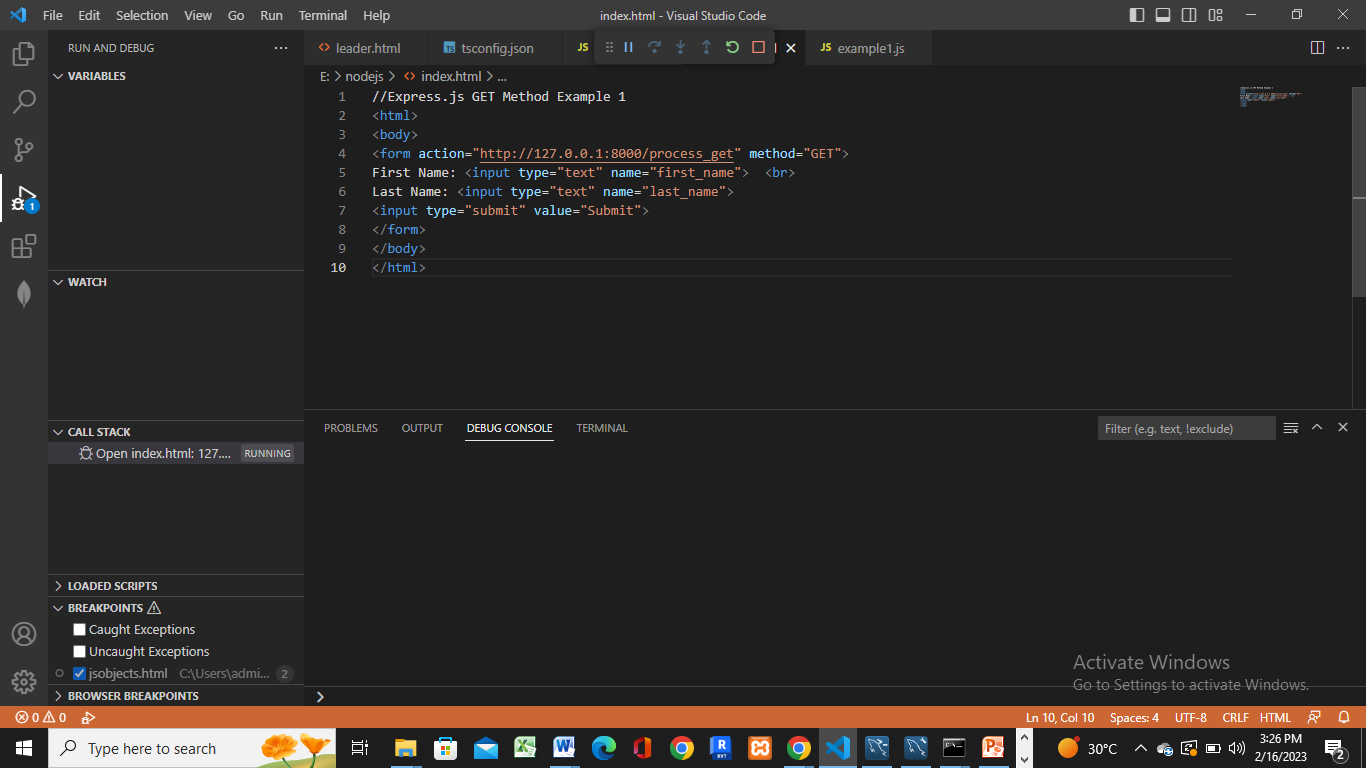


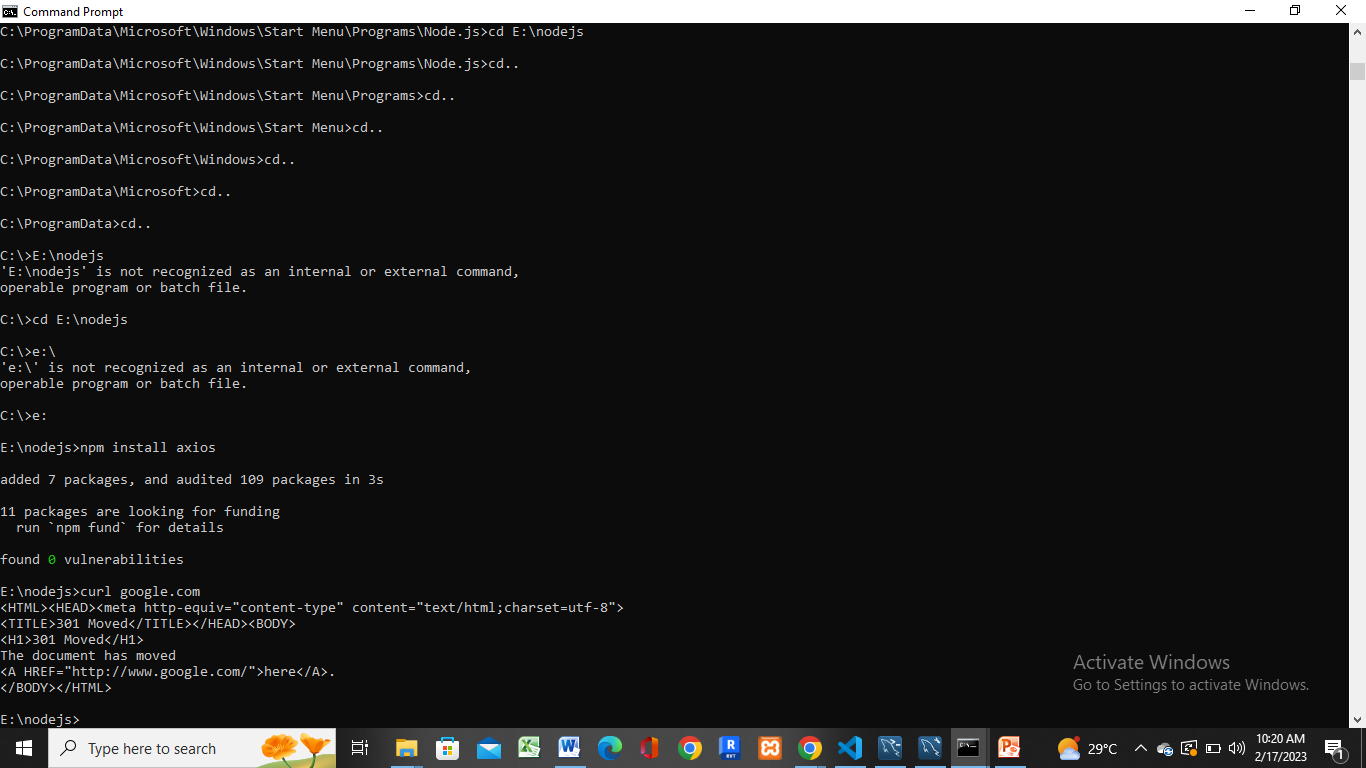


## Out put of Express.js GET Method Example 1









<http://lms.sunstone.in/mod/url/view.php?id=14> sunstone login page

<https://account.mongodb.com/account/login?n=%2Fv2%2F63eda9ee491c6a3f3186920d%23%2Fclusters> mongo db

<https://zetcode.com/javascript/axios> axios tutorial

<https://jwt.io/introduction> jwt page

<https://portal.aws.amazon.com/billing/signup?redirect=success&x-awsbc-xsrf-token=dmh6Wnc0ZERsNHlVRExseTkxc1ZBMFZJRmpTcUF6TzhhWXBOWFhwR2JVd3w3ODU3Mjk0NjA2ODUwMzEzMzUwfDF8MjAyMy0wMi0xN1QwODo1NjoxMi45NzRa&paymentProcessingSourceId=04c32f50-0f54-3253-c5c8-aa688808e671&status=REJECTED#/paymentinformation> aws page

<https://github.com/tamilinfo87/Day-4-Mango-DB/tree/main> my github

<https://www.w3schools.com/mongodb/mongodb_aggregations_intro.php> w3mongodb

<https://www.javatpoint.com/> for express js

<https://www.javatpoint.com/expressjs-file-upload> express js

C:\Tamil> cd C:\Tamil\mongosh-1.7.1-win32-x64\mongosh-1.7.1-win32-x64\bin monogsh path

E:\nodejs> node path

Microsoft Windows [Version 10.0.19044.2364]

(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>cd vue-project

The system cannot find the path specified.

C:\Users\admin>cd E:\nodejs

C:\Users\admin>cd nue project

The system cannot find the path specified.

C:\Users\admin>cd C:\Program Files\nodejs

C:\Program Files\nodejs>npm init vue@latest

Need to install the following packages:

create-vue@3.5.0

Ok to proceed? (y) y

Vue.js - The Progressive JavaScript Framework

√ Project name: ... y

√ Add TypeScript? ... No / Yes

√ Add JSX Support? ... No / Yes

√ Add Vue Router for Single Page Application development? ... No / Yes

√ Add Pinia for state management? ... No / Yes

√ Add Vitest for Unit Testing? ... No / Yes

√ Add an End-to-End Testing Solution? » No

√ Add ESLint for code quality? ... No / Yes

√ Add Prettier for code formatting? ... No / Yes

Error: EPERM: operation not permitted, mkdir 'C:\Program Files\nodejs\y'

at Object.mkdirSync (node:fs:1395:3)

at init (C:\Users\admin\AppData\Local\npm-cache\\_npx\2f7e7bff16d1c534\node\_modules\create-vue\outfile.cjs:4045:9)

at process.processTicksAndRejections (node:internal/process/task\_queues:95:5) {

errno: -4048,

syscall: 'mkdir',

code: 'EPERM',

path: 'C:\\Program Files\\nodejs\\y'

}

C:\Program Files\nodejs>npm init vue@latest

Vue.js - The Progressive JavaScript Framework

√ Project name: ... vue-project

√ Add TypeScript? ... No / Yes

√ Add JSX Support? ... No / Yes

√ Add Vue Router for Single Page Application development? ... No / Yes

√ Add Pinia for state management? ... No / Yes

√ Add Vitest for Unit Testing? ... No / Yes

√ Add an End-to-End Testing Solution? » No

√ Add ESLint for code quality? ... No / Yes

√ Add Prettier for code formatting? ... No / Yes

Error: EPERM: operation not permitted, mkdir 'C:\Program Files\nodejs\vue-project'

at Object.mkdirSync (node:fs:1395:3)

at init (C:\Users\admin\AppData\Local\npm-cache\\_npx\2f7e7bff16d1c534\node\_modules\create-vue\outfile.cjs:4045:9)

at process.processTicksAndRejections (node:internal/process/task\_queues:95:5) {

errno: -4048,

syscall: 'mkdir',

code: 'EPERM',

path: 'C:\\Program Files\\nodejs\\vue-project'

}

C:\Program Files\nodejs>dir

Volume in drive C has no label.

Volume Serial Number is 78FE-8B58

Directory of C:\Program Files\nodejs

02/14/2023 12:42 PM <DIR> .

02/14/2023 12:42 PM <DIR> ..

09/23/2022 02:40 AM 334 corepack

09/23/2022 02:40 AM 218 corepack.cmd

12/13/2022 06:59 AM 3,033 install\_tools.bat

02/01/2023 09:18 PM 69,912,696 node.exe

10/14/2021 12:30 AM 702 nodevars.bat

02/01/2023 08:38 PM 8,997 node\_etw\_provider.man

02/14/2023 12:42 PM <DIR> node\_modules

12/13/2022 06:59 AM 1,365 npm

12/13/2022 06:59 AM 483 npm.cmd

12/13/2022 06:59 AM 1,567 npx

12/13/2022 06:59 AM 539 npx.cmd

10 File(s) 69,929,934 bytes

3 Dir(s) 386,872,750,080 bytes free

C:\Program Files\nodejs>d:

The system cannot find the drive specified.

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11>cd

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11>cd..

C:\ProgramData\Microsoft\Windows\Start Menu\Programs>cd..

C:\ProgramData\Microsoft\Windows\Start Menu>cd..

C:\ProgramData\Microsoft\Windows>cd..

C:\ProgramData\Microsoft>cd..

C:\ProgramData>cd..

C:\>mkdir Jsonnew

C:\>cd Jsonnew:\

The filename, directory name, or volume label syntax is incorrect.

C:\>cd C:\Jsonnew

C:\Jsonnew>npm install json

added 1 package in 1m

C:\Jsonnew>node server.js

node:internal/modules/cjs/loader:1050

throw err;

^

Error: Cannot find module 'C:\Jsonnew\server.js'

←[90m at Module.\_resolveFilename (node:internal/modules/cjs/loader:1047:15)←[39m

←[90m at Module.\_load (node:internal/modules/cjs/loader:893:27)←[39m

←[90m at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run\_main:81:12)←[39m

←[90m at node:internal/main/run\_main\_module:23:47←[39m {

code: ←[32m'MODULE\_NOT\_FOUND'←[39m,

requireStack: []

}

Node.js v18.14.0

C:\Jsonnew>cd

C:\Jsonnew

C:\Jsonnew>cd..

C:\>cd C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules>npm install json

npm ERR! code EPERM

npm ERR! syscall mkdir

npm ERR! path C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules\json

npm ERR! errno -4048

npm ERR! Error: EPERM: operation not permitted, mkdir 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules\json'

npm ERR! [Error: EPERM: operation not permitted, mkdir 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules\json'] {

npm ERR! errno: -4048,

npm ERR! code: 'EPERM',

npm ERR! syscall: 'mkdir',

npm ERR! path: 'C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Tamil11\\node\_modules\\json'

npm ERR! }

npm ERR!

npm ERR! The operation was rejected by your operating system.

npm ERR! It's possible that the file was already in use (by a text editor or antivirus),

npm ERR! or that you lack permissions to access it.

npm ERR!

npm ERR! If you believe this might be a permissions issue, please double-check the

npm ERR! permissions of the file and its containing directories, or try running

npm ERR! the command again as root/Administrator.

npm ERR! A complete log of this run can be found in:

npm ERR! C:\Users\admin\AppData\Local\npm-cache\\_logs\2023-02-16T11\_06\_49\_854Z-debug-0.log

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules>dir

Volume in drive C has no label.

Volume Serial Number is 78FE-8B58

Directory of C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules

02/16/2023 02:44 PM <DIR> .

02/16/2023 02:44 PM <DIR> ..

02/16/2023 02:44 PM <DIR> .bin

02/16/2023 02:44 PM 21,685 .package-lock.json

02/16/2023 02:44 PM <DIR> accepts

02/16/2023 02:44 PM <DIR> array-flatten

02/16/2023 02:44 PM <DIR> body-parser

02/16/2023 02:44 PM <DIR> bytes

02/16/2023 02:44 PM <DIR> call-bind

02/16/2023 02:44 PM <DIR> content-disposition

02/16/2023 02:44 PM <DIR> content-type

02/16/2023 02:44 PM <DIR> cookie

02/16/2023 02:44 PM <DIR> cookie-signature

02/16/2023 02:44 PM <DIR> debug

02/16/2023 02:44 PM <DIR> depd

02/16/2023 02:44 PM <DIR> destroy

02/16/2023 02:44 PM <DIR> ee-first

02/16/2023 02:44 PM <DIR> encodeurl

02/16/2023 02:44 PM <DIR> escape-html

02/16/2023 02:44 PM <DIR> etag

02/16/2023 02:44 PM <DIR> express

02/16/2023 02:44 PM <DIR> finalhandler

02/16/2023 02:44 PM <DIR> forwarded

02/16/2023 02:44 PM <DIR> fresh

02/16/2023 02:44 PM <DIR> function-bind

02/16/2023 02:44 PM <DIR> get-intrinsic

02/16/2023 02:44 PM <DIR> has

02/16/2023 02:44 PM <DIR> has-symbols

02/16/2023 02:44 PM <DIR> http-errors

02/16/2023 02:44 PM <DIR> iconv-lite

02/16/2023 02:44 PM <DIR> inherits

02/16/2023 02:44 PM <DIR> ipaddr.js

02/16/2023 02:44 PM <DIR> media-typer

02/16/2023 02:44 PM <DIR> merge-descriptors

02/16/2023 02:44 PM <DIR> methods

02/16/2023 02:44 PM <DIR> mime

02/16/2023 02:44 PM <DIR> mime-db

02/16/2023 02:44 PM <DIR> mime-types

02/16/2023 02:44 PM <DIR> ms

02/16/2023 02:44 PM <DIR> negotiator

02/16/2023 02:44 PM <DIR> object-inspect

02/16/2023 02:44 PM <DIR> on-finished

02/16/2023 02:44 PM <DIR> parseurl

02/16/2023 02:44 PM <DIR> path-to-regexp

02/16/2023 02:44 PM <DIR> proxy-addr

02/16/2023 02:44 PM <DIR> qs

02/16/2023 02:44 PM <DIR> range-parser

02/16/2023 02:44 PM <DIR> raw-body

02/16/2023 02:44 PM <DIR> safe-buffer

02/16/2023 02:44 PM <DIR> safer-buffer

02/16/2023 02:44 PM <DIR> send

02/16/2023 02:44 PM <DIR> serve-static

02/16/2023 02:44 PM <DIR> setprototypeof

02/16/2023 02:44 PM <DIR> side-channel

02/16/2023 02:44 PM <DIR> statuses

02/16/2023 02:44 PM <DIR> toidentifier

02/16/2023 02:44 PM <DIR> type-is

02/16/2023 02:44 PM <DIR> unpipe

02/16/2023 02:44 PM <DIR> utils-merge

02/16/2023 02:44 PM <DIR> vary

1 File(s) 21,685 bytes

59 Dir(s) 383,142,887,424 bytes free

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules>cd..

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11>cd..

C:\ProgramData\Microsoft\Windows\Start Menu\Programs>cd..

C:\ProgramData\Microsoft\Windows\Start Menu>cd..

C:\ProgramData\Microsoft\Windows>cd..

C:\ProgramData\Microsoft>cd..

C:\ProgramData>cd..

C:\>cd C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Node.js

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Node.js>mk dir Tamilj

'mk' is not recognized as an internal or external command,

operable program or batch file.

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Node.js>mkdir Tamilj

Access is denied.

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Node.js>cd C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules>json -v

'json' is not recognized as an internal or external command,

operable program or batch file.

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules>jason -v

'jason' is not recognized as an internal or external command,

operable program or batch file.

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules>dir

Volume in drive C has no label.

Volume Serial Number is 78FE-8B58

Directory of C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules

02/16/2023 02:44 PM <DIR> .

02/16/2023 02:44 PM <DIR> ..

02/16/2023 02:44 PM <DIR> .bin

02/16/2023 02:44 PM 21,685 .package-lock.json

02/16/2023 02:44 PM <DIR> accepts

02/16/2023 02:44 PM <DIR> array-flatten

02/16/2023 02:44 PM <DIR> body-parser

02/16/2023 02:44 PM <DIR> bytes

02/16/2023 02:44 PM <DIR> call-bind

02/16/2023 02:44 PM <DIR> content-disposition

02/16/2023 02:44 PM <DIR> content-type

02/16/2023 02:44 PM <DIR> cookie

02/16/2023 02:44 PM <DIR> cookie-signature

02/16/2023 02:44 PM <DIR> debug

02/16/2023 02:44 PM <DIR> depd

02/16/2023 02:44 PM <DIR> destroy

02/16/2023 02:44 PM <DIR> ee-first

02/16/2023 02:44 PM <DIR> encodeurl

02/16/2023 02:44 PM <DIR> escape-html

02/16/2023 02:44 PM <DIR> etag

02/16/2023 02:44 PM <DIR> express

02/16/2023 02:44 PM <DIR> finalhandler

02/16/2023 02:44 PM <DIR> forwarded

02/16/2023 02:44 PM <DIR> fresh

02/16/2023 02:44 PM <DIR> function-bind

02/16/2023 02:44 PM <DIR> get-intrinsic

02/16/2023 02:44 PM <DIR> has

02/16/2023 02:44 PM <DIR> has-symbols

02/16/2023 02:44 PM <DIR> http-errors

02/16/2023 02:44 PM <DIR> iconv-lite

02/16/2023 02:44 PM <DIR> inherits

02/16/2023 02:44 PM <DIR> ipaddr.js

02/16/2023 02:44 PM <DIR> media-typer

02/16/2023 02:44 PM <DIR> merge-descriptors

02/16/2023 02:44 PM <DIR> methods

02/16/2023 02:44 PM <DIR> mime

02/16/2023 02:44 PM <DIR> mime-db

02/16/2023 02:44 PM <DIR> mime-types

02/16/2023 02:44 PM <DIR> ms

02/16/2023 02:44 PM <DIR> negotiator

02/16/2023 02:44 PM <DIR> object-inspect

02/16/2023 02:44 PM <DIR> on-finished

02/16/2023 02:44 PM <DIR> parseurl

02/16/2023 02:44 PM <DIR> path-to-regexp

02/16/2023 02:44 PM <DIR> proxy-addr

02/16/2023 02:44 PM <DIR> qs

02/16/2023 02:44 PM <DIR> range-parser

02/16/2023 02:44 PM <DIR> raw-body

02/16/2023 02:44 PM <DIR> safe-buffer

02/16/2023 02:44 PM <DIR> safer-buffer

02/16/2023 02:44 PM <DIR> send

02/16/2023 02:44 PM <DIR> serve-static

02/16/2023 02:44 PM <DIR> setprototypeof

02/16/2023 02:44 PM <DIR> side-channel

02/16/2023 02:44 PM <DIR> statuses

02/16/2023 02:44 PM <DIR> toidentifier

02/16/2023 02:44 PM <DIR> type-is

02/16/2023 02:44 PM <DIR> unpipe

02/16/2023 02:44 PM <DIR> utils-merge

02/16/2023 02:44 PM <DIR> vary

1 File(s) 21,685 bytes

59 Dir(s) 383,191,457,792 bytes free

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules>json -v

'json' is not recognized as an internal or external command,

operable program or batch file.

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules>cd C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11>node express1.js

Example app listening at http://:::8000

^C

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11>node -v

v18.14.0

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11>json -v

'json' is not recognized as an internal or external command,

operable program or batch file.

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11>npm install axios

npm ERR! code EPERM

npm ERR! syscall mkdir

npm ERR! path C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules\axios

npm ERR! errno -4048

npm ERR! Error: EPERM: operation not permitted, mkdir 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules\axios'

npm ERR! [Error: EPERM: operation not permitted, mkdir 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11\node\_modules\axios'] {

npm ERR! errno: -4048,

npm ERR! code: 'EPERM',

npm ERR! syscall: 'mkdir',

npm ERR! path: 'C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Tamil11\\node\_modules\\axios'

npm ERR! }

npm ERR!

npm ERR! The operation was rejected by your operating system.

npm ERR! It's possible that the file was already in use (by a text editor or antivirus),

npm ERR! or that you lack permissions to access it.

npm ERR!

npm ERR! If you believe this might be a permissions issue, please double-check the

npm ERR! permissions of the file and its containing directories, or try running

npm ERR! the command again as root/Administrator.

npm ERR! A complete log of this run can be found in:

npm ERR! C:\Users\admin\AppData\Local\npm-cache\\_logs\2023-02-17T04\_46\_36\_370Z-debug-0.log

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Tamil11>cd C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Node.js

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Node.js>npm install axios

npm ERR! code EPERM

npm ERR! syscall mkdir

npm ERR! path C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Node.js\node\_modules

npm ERR! errno -4048

npm ERR! Error: EPERM: operation not permitted, mkdir 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Node.js\node\_modules'

npm ERR! [Error: EPERM: operation not permitted, mkdir 'C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Node.js\node\_modules'] {

npm ERR! errno: -4048,

npm ERR! code: 'EPERM',

npm ERR! syscall: 'mkdir',

npm ERR! path: 'C:\\ProgramData\\Microsoft\\Windows\\Start Menu\\Programs\\Node.js\\node\_modules'

npm ERR! }

npm ERR!

npm ERR! The operation was rejected by your operating system.

npm ERR! It's possible that the file was already in use (by a text editor or antivirus),

npm ERR! or that you lack permissions to access it.

npm ERR!

npm ERR! If you believe this might be a permissions issue, please double-check the

npm ERR! permissions of the file and its containing directories, or try running

npm ERR! the command again as root/Administrator.

npm ERR! A complete log of this run can be found in:

npm ERR! C:\Users\admin\AppData\Local\npm-cache\\_logs\2023-02-17T04\_47\_44\_508Z-debug-0.log

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Node.js>cd E:\nodejs

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Node.js>cd..

C:\ProgramData\Microsoft\Windows\Start Menu\Programs>cd..

C:\ProgramData\Microsoft\Windows\Start Menu>cd..

C:\ProgramData\Microsoft\Windows>cd..

C:\ProgramData\Microsoft>cd..

C:\ProgramData>cd..

C:\>E:\nodejs

'E:\nodejs' is not recognized as an internal or external command,

operable program or batch file.

C:\>cd E:\nodejs

C:\>e:\

'e:\' is not recognized as an internal or external command,

operable program or batch file.

C:\>e:

E:\nodejs>npm install axios

added 7 packages, and audited 109 packages in 3s

11 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

E:\nodejs>curl google.com

<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">

<TITLE>301 Moved</TITLE></HEAD><BODY>

<H1>301 Moved</H1>

The document has moved

<A HREF="http://www.google.com/">here</A>.

</BODY></HTML>

E:\nodejs>dir

Volume in drive E has no label.

Volume Serial Number is D440-B9AA

Directory of E:\nodejs

02/17/2023 10:21 AM <DIR> .

02/17/2023 10:21 AM <DIR> ..

02/15/2023 02:47 PM 73 app.js

02/15/2023 03:21 PM 81 app.ts

02/15/2023 12:40 PM 161 app2.vue

02/17/2023 10:21 AM 122 axios1.js

02/15/2023 10:40 AM 5,672 bird150.jpg

02/15/2023 10:17 AM 1,786 bootclounm.html

02/15/2023 11:05 AM 1,779 bootclounmimage.html

02/15/2023 10:13 AM 1,391 bootstrap.html

02/15/2023 10:45 AM 1,556 boottheem2.html

02/15/2023 10:44 AM 669 boottheme.html

02/15/2023 10:39 AM 1,788 carousel.html

02/15/2023 10:37 AM 71,294 chicago.jpg

02/15/2023 10:16 AM 89,682 cinqueterre.jpg

02/17/2023 09:59 AM 652 example1.js

02/16/2023 03:48 PM 520 example2.js

02/16/2023 03:26 PM 302 index.html

02/16/2023 03:38 PM 784 index1.html

02/16/2023 04:22 PM 1,221 index3.html

02/15/2023 10:36 AM 22,914 la.jpg

02/15/2023 11:39 AM 999 leader.html

02/14/2023 03:21 PM 185 main1.js

02/14/2023 03:11 PM 56 module1.js

02/14/2023 03:13 PM 549 module2.js

02/14/2023 03:21 PM 14 mynewfile.txt

02/15/2023 10:29 AM 33,636 newyork.jpg

02/14/2023 12:51 PM 175 node1.js

02/14/2023 01:00 PM 440 node11.js

02/14/2023 02:42 PM 457 node2.js

02/14/2023 03:23 PM 643 node3.js

02/17/2023 10:19 AM <DIR> node\_modules

02/15/2023 10:37 AM 53,292 ny.jpg

02/17/2023 10:19 AM 41,467 package-lock.json

02/17/2023 10:19 AM 291 package.json

02/16/2023 04:21 PM 260 package1.json

02/16/2023 04:13 PM 1,066 routing\_example.js

02/15/2023 10:29 AM 16,102 sanfran.jpg

02/16/2023 04:24 PM 849 server.js

02/15/2023 11:12 AM 5,415 sw.jpeg

02/15/2023 11:39 AM 766 threeimgs.html

02/15/2023 02:30 PM 11,401 tsconfig.json

02/15/2023 11:54 AM <DIR> y

39 File(s) 370,510 bytes

4 Dir(s) 471,842,242,560 bytes free

E:\nodejs>cd express

The system cannot find the path specified.

E:\nodejs>cd C:\Tamil

E:\nodejs>cd..

E:\>cd C:\Tamil

E:\>cd..

E:\>C:\Tamil

'C:\Tamil' is not recognized as an internal or external command,

operable program or batch file.

E:\>cd C:\Tamil

E:\>c:

C:\Tamil>node axios1.js

node:internal/modules/cjs/loader:1050

throw err;

^

Error: Cannot find module 'axios'

Require stack:

- C:\Tamil\axios1.js

←[90m at Module.\_resolveFilename (node:internal/modules/cjs/loader:1047:15)←[39m

←[90m at Module.\_load (node:internal/modules/cjs/loader:893:27)←[39m

←[90m at Module.require (node:internal/modules/cjs/loader:1113:19)←[39m

←[90m at require (node:internal/modules/cjs/helpers:103:18)←[39m

at Object.<anonymous> ←[90m(C:\Tamil\←[39maxios1.js:1:15←[90m)←[39m

←[90m at Module.\_compile (node:internal/modules/cjs/loader:1226:14)←[39m

←[90m at Module.\_extensions..js (node:internal/modules/cjs/loader:1280:10)←[39m

←[90m at Module.load (node:internal/modules/cjs/loader:1089:32)←[39m

←[90m at Module.\_load (node:internal/modules/cjs/loader:930:12)←[39m

←[90m at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run\_main:81:12)←[39m {

code: ←[32m'MODULE\_NOT\_FOUND'←[39m,

requireStack: [ ←[32m'C:\\Tamil\\axios1.js'←[39m ]

}

Node.js v18.14.0

C:\Tamil>cd..

C:\>cd E:\nodejs

C:\>E:\

'E:\' is not recognized as an internal or external command,

operable program or batch file.

C:\>e:

E:\nodejs>node axios1.js

E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825

AxiosError.call(axiosError, error.message, code, config, request, response);

^

AxiosError: connect ECONNREFUSED ::1:5000

at AxiosError.from (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825:14)

at RedirectableRequest.handleRequestError (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:2965:25)

at RedirectableRequest.emit (node:events:513:28)

at eventHandlers.<computed> (E:\nodejs\node\_modules\follow-redirects\index.js:14:24)

at ClientRequest.emit (node:events:513:28)

at Socket.socketErrorListener (node:\_http\_client:502:9)

at Socket.emit (node:events:513:28)

at emitErrorNT (node:internal/streams/destroy:151:8)

at emitErrorCloseNT (node:internal/streams/destroy:116:3)

at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) {

port: 5000,

address: '::1',

syscall: 'connect',

code: 'ECONNREFUSED',

errno: -4078,

config: {

transitional: {

silentJSONParsing: true,

forcedJSONParsing: true,

clarifyTimeoutError: false

},

adapter: [ 'xhr', 'http' ],

transformRequest: [ [Function: transformRequest] ],

transformResponse: [ [Function: transformResponse] ],

timeout: 0,

xsrfCookieName: 'XSRF-TOKEN',

xsrfHeaderName: 'X-XSRF-TOKEN',

maxContentLength: -1,

maxBodyLength: -1,

env: {

FormData: [Function: FormData] {

LINE\_BREAK: '\r\n',

DEFAULT\_CONTENT\_TYPE: 'application/octet-stream'

},

Blob: [class Blob]

},

validateStatus: [Function: validateStatus],

headers: AxiosHeaders {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

method: 'post',

url: 'http://localhost:5000/',

data: undefined

},

request: <ref \*2> Writable {

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: false,

decodeStrings: true,

defaultEncoding: 'utf8',

length: 0,

writing: false,

corked: 0,

sync: true,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: null,

writelen: 0,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 0,

constructed: true,

prefinished: false,

errorEmitted: false,

emitClose: true,

autoDestroy: true,

errored: null,

closed: false,

closeEmitted: false,

[Symbol(kOnFinished)]: []

},

\_events: [Object: null prototype] {

response: [Function: handleResponse],

error: [Function: handleRequestError],

socket: [Function: handleRequestSocket]

},

\_eventsCount: 3,

\_maxListeners: undefined,

\_options: {

maxRedirects: 21,

maxBodyLength: Infinity,

protocol: 'http:',

path: '/',

method: 'POST',

headers: [Object: null prototype] {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

agents: { http: undefined, https: undefined },

auth: undefined,

beforeRedirect: [Function: dispatchBeforeRedirect],

beforeRedirects: { proxy: [Function: beforeRedirect] },

hostname: 'localhost',

port: '5000',

agent: undefined,

nativeProtocols: {

'http:': {

\_connectionListener: [Function: connectionListener],

METHODS: [

'ACL', 'BIND', 'CHECKOUT',

'CONNECT', 'COPY', 'DELETE',

'GET', 'HEAD', 'LINK',

'LOCK', 'M-SEARCH', 'MERGE',

'MKACTIVITY', 'MKCALENDAR', 'MKCOL',

'MOVE', 'NOTIFY', 'OPTIONS',

'PATCH', 'POST', 'PROPFIND',

'PROPPATCH', 'PURGE', 'PUT',

'REBIND', 'REPORT', 'SEARCH',

'SOURCE', 'SUBSCRIBE', 'TRACE',

'UNBIND', 'UNLINK', 'UNLOCK',

'UNSUBSCRIBE'

],

STATUS\_CODES: {

'100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'103': 'Early Hints',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'208': 'Already Reported',

'226': 'IM Used',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Found',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'308': 'Permanent Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Timeout',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Payload Too Large',

'414': 'URI Too Long',

'415': 'Unsupported Media Type',

'416': 'Range Not Satisfiable',

'417': 'Expectation Failed',

'418': "I'm a Teapot",

'421': 'Misdirected Request',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Too Early',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'451': 'Unavailable For Legal Reasons',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Timeout',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'508': 'Loop Detected',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required'

},

Agent: [Function: Agent] { defaultMaxSockets: Infinity },

ClientRequest: [Function: ClientRequest],

IncomingMessage: [Function: IncomingMessage],

OutgoingMessage: [Function: OutgoingMessage],

Server: [Function: Server],

ServerResponse: [Function: ServerResponse],

createServer: [Function: createServer],

validateHeaderName: [Function: \_\_node\_internal\_],

validateHeaderValue: [Function: \_\_node\_internal\_],

get: [Function: get],

request: [Function: request],

setMaxIdleHTTPParsers: [Function: setMaxIdleHTTPParsers],

maxHeaderSize: [Getter],

globalAgent: [Getter/Setter]

},

'https:': {

Agent: [Function: Agent],

globalAgent: Agent {

\_events: [Object: null prototype],

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 443,

protocol: 'https:',

options: [Object: null prototype],

requests: [Object: null prototype] {},

sockets: [Object: null prototype] {},

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 0,

maxCachedSessions: 100,

\_sessionCache: [Object],

[Symbol(kCapture)]: false

},

Server: [Function: Server],

createServer: [Function: createServer],

get: [Function: get],

request: [Function: request]

}

},

pathname: '/'

},

\_ended: true,

\_ending: true,

\_redirectCount: 0,

\_redirects: [],

\_requestBodyLength: 0,

\_requestBodyBuffers: [],

\_onNativeResponse: [Function (anonymous)],

\_currentRequest: <ref \*1> ClientRequest {

\_events: [Object: null prototype] {

response: [Function: bound onceWrapper] {

listener: [Function (anonymous)]

},

abort: [Function (anonymous)],

aborted: [Function (anonymous)],

connect: [Function (anonymous)],

error: [Function (anonymous)],

socket: [Function (anonymous)],

timeout: [Function (anonymous)]

},

\_eventsCount: 7,

\_maxListeners: undefined,

outputData: [],

outputSize: 0,

writable: true,

destroyed: false,

\_last: true,

chunkedEncoding: false,

shouldKeepAlive: false,

maxRequestsOnConnectionReached: false,

\_defaultKeepAlive: true,

useChunkedEncodingByDefault: true,

sendDate: false,

\_removedConnection: false,

\_removedContLen: false,

\_removedTE: false,

strictContentLength: false,

\_contentLength: 0,

\_hasBody: true,

\_trailer: '',

finished: true,

\_headerSent: true,

\_closed: false,

socket: Socket {

connecting: false,

\_hadError: true,

\_parent: null,

\_host: 'localhost',

\_closeAfterHandlingError: false,

\_readableState: ReadableState {

objectMode: false,

highWaterMark: 16384,

buffer: BufferList { head: null, tail: null, length: 0 },

length: 0,

pipes: [],

flowing: true,

ended: false,

endEmitted: false,

reading: true,

constructed: true,

sync: false,

needReadable: true,

emittedReadable: false,

readableListening: false,

resumeScheduled: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

destroyed: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

defaultEncoding: 'utf8',

awaitDrainWriters: null,

multiAwaitDrain: false,

readingMore: false,

dataEmitted: false,

decoder: null,

encoding: null,

[Symbol(kPaused)]: false

},

\_events: [Object: null prototype] {

end: [Function: onReadableStreamEnd],

connect: [ [Function], [Function], [Function] ],

free: [Function: onFree],

close: [

[Function: onClose],

[Function: socketCloseListener],

[Function]

],

timeout: [Function: onTimeout],

agentRemove: [Function: onRemove],

error: [Function: socketErrorListener],

drain: [Function: ondrain]

},

\_eventsCount: 8,

\_maxListeners: undefined,

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: true,

decodeStrings: false,

defaultEncoding: 'utf8',

length: 242,

writing: true,

corked: 0,

sync: false,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: [Function: bound onFinish],

writelen: 242,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 1,

constructed: true,

prefinished: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

[Symbol(kOnFinished)]: []

},

allowHalfOpen: false,

\_sockname: null,

\_pendingData: 'POST / HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_pendingEncoding: 'latin1',

server: null,

\_server: null,

parser: null,

\_httpMessage: [Circular \*1],

[Symbol(async\_id\_symbol)]: 4,

[Symbol(kHandle)]: null,

[Symbol(lastWriteQueueSize)]: 0,

[Symbol(timeout)]: null,

[Symbol(kBuffer)]: null,

[Symbol(kBufferCb)]: null,

[Symbol(kBufferGen)]: null,

[Symbol(kCapture)]: false,

[Symbol(kSetNoDelay)]: true,

[Symbol(kSetKeepAlive)]: true,

[Symbol(kSetKeepAliveInitialDelay)]: 60,

[Symbol(kBytesRead)]: 0,

[Symbol(kBytesWritten)]: 0

},

\_header: 'POST / HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_keepAliveTimeout: 0,

\_onPendingData: [Function: nop],

agent: Agent {

\_events: [Object: null prototype] {

free: [Function (anonymous)],

newListener: [Function: maybeEnableKeylog]

},

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 80,

protocol: 'http:',

options: [Object: null prototype] { noDelay: true, path: null },

requests: [Object: null prototype] {},

sockets: [Object: null prototype] { 'localhost:5000:': [ [Socket] ] },

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 1,

[Symbol(kCapture)]: false

},

socketPath: undefined,

method: 'POST',

maxHeaderSize: undefined,

insecureHTTPParser: undefined,

joinDuplicateHeaders: undefined,

path: '/',

\_ended: false,

res: null,

aborted: false,

timeoutCb: null,

upgradeOrConnect: false,

parser: null,

maxHeadersCount: null,

reusedSocket: false,

host: 'localhost',

protocol: 'http:',

\_redirectable: [Circular \*2],

[Symbol(kCapture)]: false,

[Symbol(kBytesWritten)]: 0,

[Symbol(kEndCalled)]: true,

[Symbol(kNeedDrain)]: false,

[Symbol(corked)]: 0,

[Symbol(kOutHeaders)]: [Object: null prototype] {

accept: [ 'Accept', 'application/json, text/plain, \*/\*' ],

'content-type': [ 'Content-Type', 'application/x-www-form-urlencoded' ],

'user-agent': [ 'User-Agent', 'axios/1.3.3' ],

'accept-encoding': [ 'Accept-Encoding', 'gzip, compress, deflate, br' ],

host: [ 'Host', 'localhost:5000' ]

},

[Symbol(errored)]: null,

[Symbol(kUniqueHeaders)]: null

},

\_currentUrl: 'http://localhost:5000/',

[Symbol(kCapture)]: false

},

cause: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

}

}

Node.js v18.14.0

E:\nodejs>node axios2.js

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="format.css">

<title>My html page</title>

</head>

<body>

<p>

Today is a beautiful day. We go swimming and fishing.

</p>

<p>

Hello there. How are you?

</p>

</body>

</html>

E:\nodejs>

E:\nodejs>

E:\nodejs>npm install express

up to date, audited 109 packages in 1s

11 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

E:\nodejs>npm install doten

npm ERR! code E404

npm ERR! 404 Not Found - GET https://registry.npmjs.org/doten - Not found

npm ERR! 404

npm ERR! 404 'doten@\*' is not in this registry.

npm ERR! 404

npm ERR! 404 Note that you can also install from a

npm ERR! 404 tarball, folder, http url, or git url.

npm ERR! A complete log of this run can be found in:

npm ERR! C:\Users\admin\AppData\Local\npm-cache\\_logs\2023-02-17T05\_16\_30\_370Z-debug-0.log

E:\nodejs>npm install dotenv

added 1 package, and audited 110 packages in 1s

11 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

E:\nodejs>npm install jsonwebtoken

added 10 packages, and audited 120 packages in 7s

11 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

E:\nodejs>npm init -y

Wrote to E:\nodejs\package.json:

{

"dependencies": {

"axios": "^1.3.3",

"dotenv": "^16.0.3",

"express": "^4.18.2",

"formidable": "^2.1.1",

"fs": "^0.0.1-security",

"http": "^0.0.1-security",

"jsonwebtoken": "^9.0.0",

"mysql": "^2.18.1",

"vu": "^0.0.0",

"vue-router": "^4.1.6"

},

"devDependencies": {

"typescript": "^4.9.5"

},

"scripts": {

"start": "node server.js"

},

"name": "nodejs",

"version": "1.0.0",

"main": "app.js",

"keywords": [],

"author": "",

"license": "ISC",

"description": ""

}

E:\nodejs>create a file .env

'create' is not recognized as an internal or external command,

operable program or batch file.

E:\nodejs>create a Tamil.env

'create' is not recognized as an internal or external command,

operable program or batch file.

E:\nodejs>node index4.js

Server is up and running on 5000 ...

^C

E:\nodejs>^D

E:\nodejs>node axios3.js

E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825

AxiosError.call(axiosError, error.message, code, config, request, response);

^

AxiosError: connect ECONNREFUSED ::1:5000

at AxiosError.from (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825:14)

at RedirectableRequest.handleRequestError (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:2965:25)

at RedirectableRequest.emit (node:events:513:28)

at eventHandlers.<computed> (E:\nodejs\node\_modules\follow-redirects\index.js:14:24)

at ClientRequest.emit (node:events:513:28)

at Socket.socketErrorListener (node:\_http\_client:502:9)

at Socket.emit (node:events:513:28)

at emitErrorNT (node:internal/streams/destroy:151:8)

at emitErrorCloseNT (node:internal/streams/destroy:116:3)

at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) {

port: 5000,

address: '::1',

syscall: 'connect',

code: 'ECONNREFUSED',

errno: -4078,

config: {

transitional: {

silentJSONParsing: true,

forcedJSONParsing: true,

clarifyTimeoutError: false

},

adapter: [ 'xhr', 'http' ],

transformRequest: [ [Function: transformRequest] ],

transformResponse: [ [Function: transformResponse] ],

timeout: 0,

xsrfCookieName: 'XSRF-TOKEN',

xsrfHeaderName: 'X-XSRF-TOKEN',

maxContentLength: -1,

maxBodyLength: -1,

env: {

FormData: [Function: FormData] {

LINE\_BREAK: '\r\n',

DEFAULT\_CONTENT\_TYPE: 'application/octet-stream'

},

Blob: [class Blob]

},

validateStatus: [Function: validateStatus],

headers: AxiosHeaders {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

method: 'post',

url: 'http://localhost:5000/user/generateToken',

data: undefined

},

request: <ref \*2> Writable {

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: false,

decodeStrings: true,

defaultEncoding: 'utf8',

length: 0,

writing: false,

corked: 0,

sync: true,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: null,

writelen: 0,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 0,

constructed: true,

prefinished: false,

errorEmitted: false,

emitClose: true,

autoDestroy: true,

errored: null,

closed: false,

closeEmitted: false,

[Symbol(kOnFinished)]: []

},

\_events: [Object: null prototype] {

response: [Function: handleResponse],

error: [Function: handleRequestError],

socket: [Function: handleRequestSocket]

},

\_eventsCount: 3,

\_maxListeners: undefined,

\_options: {

maxRedirects: 21,

maxBodyLength: Infinity,

protocol: 'http:',

path: '/user/generateToken',

method: 'POST',

headers: [Object: null prototype] {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

agents: { http: undefined, https: undefined },

auth: undefined,

beforeRedirect: [Function: dispatchBeforeRedirect],

beforeRedirects: { proxy: [Function: beforeRedirect] },

hostname: 'localhost',

port: '5000',

agent: undefined,

nativeProtocols: {

'http:': {

\_connectionListener: [Function: connectionListener],

METHODS: [

'ACL', 'BIND', 'CHECKOUT',

'CONNECT', 'COPY', 'DELETE',

'GET', 'HEAD', 'LINK',

'LOCK', 'M-SEARCH', 'MERGE',

'MKACTIVITY', 'MKCALENDAR', 'MKCOL',

'MOVE', 'NOTIFY', 'OPTIONS',

'PATCH', 'POST', 'PROPFIND',

'PROPPATCH', 'PURGE', 'PUT',

'REBIND', 'REPORT', 'SEARCH',

'SOURCE', 'SUBSCRIBE', 'TRACE',

'UNBIND', 'UNLINK', 'UNLOCK',

'UNSUBSCRIBE'

],

STATUS\_CODES: {

'100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'103': 'Early Hints',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'208': 'Already Reported',

'226': 'IM Used',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Found',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'308': 'Permanent Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Timeout',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Payload Too Large',

'414': 'URI Too Long',

'415': 'Unsupported Media Type',

'416': 'Range Not Satisfiable',

'417': 'Expectation Failed',

'418': "I'm a Teapot",

'421': 'Misdirected Request',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Too Early',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'451': 'Unavailable For Legal Reasons',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Timeout',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'508': 'Loop Detected',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required'

},

Agent: [Function: Agent] { defaultMaxSockets: Infinity },

ClientRequest: [Function: ClientRequest],

IncomingMessage: [Function: IncomingMessage],

OutgoingMessage: [Function: OutgoingMessage],

Server: [Function: Server],

ServerResponse: [Function: ServerResponse],

createServer: [Function: createServer],

validateHeaderName: [Function: \_\_node\_internal\_],

validateHeaderValue: [Function: \_\_node\_internal\_],

get: [Function: get],

request: [Function: request],

setMaxIdleHTTPParsers: [Function: setMaxIdleHTTPParsers],

maxHeaderSize: [Getter],

globalAgent: [Getter/Setter]

},

'https:': {

Agent: [Function: Agent],

globalAgent: Agent {

\_events: [Object: null prototype],

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 443,

protocol: 'https:',

options: [Object: null prototype],

requests: [Object: null prototype] {},

sockets: [Object: null prototype] {},

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 0,

maxCachedSessions: 100,

\_sessionCache: [Object],

[Symbol(kCapture)]: false

},

Server: [Function: Server],

createServer: [Function: createServer],

get: [Function: get],

request: [Function: request]

}

},

pathname: '/user/generateToken'

},

\_ended: true,

\_ending: true,

\_redirectCount: 0,

\_redirects: [],

\_requestBodyLength: 0,

\_requestBodyBuffers: [],

\_onNativeResponse: [Function (anonymous)],

\_currentRequest: <ref \*1> ClientRequest {

\_events: [Object: null prototype] {

response: [Function: bound onceWrapper] {

listener: [Function (anonymous)]

},

abort: [Function (anonymous)],

aborted: [Function (anonymous)],

connect: [Function (anonymous)],

error: [Function (anonymous)],

socket: [Function (anonymous)],

timeout: [Function (anonymous)]

},

\_eventsCount: 7,

\_maxListeners: undefined,

outputData: [],

outputSize: 0,

writable: true,

destroyed: false,

\_last: true,

chunkedEncoding: false,

shouldKeepAlive: false,

maxRequestsOnConnectionReached: false,

\_defaultKeepAlive: true,

useChunkedEncodingByDefault: true,

sendDate: false,

\_removedConnection: false,

\_removedContLen: false,

\_removedTE: false,

strictContentLength: false,

\_contentLength: 0,

\_hasBody: true,

\_trailer: '',

finished: true,

\_headerSent: true,

\_closed: false,

socket: Socket {

connecting: false,

\_hadError: true,

\_parent: null,

\_host: 'localhost',

\_closeAfterHandlingError: false,

\_readableState: ReadableState {

objectMode: false,

highWaterMark: 16384,

buffer: BufferList { head: null, tail: null, length: 0 },

length: 0,

pipes: [],

flowing: true,

ended: false,

endEmitted: false,

reading: true,

constructed: true,

sync: false,

needReadable: true,

emittedReadable: false,

readableListening: false,

resumeScheduled: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

destroyed: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

defaultEncoding: 'utf8',

awaitDrainWriters: null,

multiAwaitDrain: false,

readingMore: false,

dataEmitted: false,

decoder: null,

encoding: null,

[Symbol(kPaused)]: false

},

\_events: [Object: null prototype] {

end: [Function: onReadableStreamEnd],

connect: [ [Function], [Function], [Function] ],

free: [Function: onFree],

close: [

[Function: onClose],

[Function: socketCloseListener],

[Function]

],

timeout: [Function: onTimeout],

agentRemove: [Function: onRemove],

error: [Function: socketErrorListener],

drain: [Function: ondrain]

},

\_eventsCount: 8,

\_maxListeners: undefined,

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: true,

decodeStrings: false,

defaultEncoding: 'utf8',

length: 260,

writing: true,

corked: 0,

sync: false,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: [Function: bound onFinish],

writelen: 260,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 1,

constructed: true,

prefinished: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

[Symbol(kOnFinished)]: []

},

allowHalfOpen: false,

\_sockname: null,

\_pendingData: 'POST /user/generateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_pendingEncoding: 'latin1',

server: null,

\_server: null,

parser: null,

\_httpMessage: [Circular \*1],

[Symbol(async\_id\_symbol)]: 4,

[Symbol(kHandle)]: null,

[Symbol(lastWriteQueueSize)]: 0,

[Symbol(timeout)]: null,

[Symbol(kBuffer)]: null,

[Symbol(kBufferCb)]: null,

[Symbol(kBufferGen)]: null,

[Symbol(kCapture)]: false,

[Symbol(kSetNoDelay)]: true,

[Symbol(kSetKeepAlive)]: true,

[Symbol(kSetKeepAliveInitialDelay)]: 60,

[Symbol(kBytesRead)]: 0,

[Symbol(kBytesWritten)]: 0

},

\_header: 'POST /user/generateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_keepAliveTimeout: 0,

\_onPendingData: [Function: nop],

agent: Agent {

\_events: [Object: null prototype] {

free: [Function (anonymous)],

newListener: [Function: maybeEnableKeylog]

},

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 80,

protocol: 'http:',

options: [Object: null prototype] { noDelay: true, path: null },

requests: [Object: null prototype] {},

sockets: [Object: null prototype] { 'localhost:5000:': [ [Socket] ] },

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 1,

[Symbol(kCapture)]: false

},

socketPath: undefined,

method: 'POST',

maxHeaderSize: undefined,

insecureHTTPParser: undefined,

joinDuplicateHeaders: undefined,

path: '/user/generateToken',

\_ended: false,

res: null,

aborted: false,

timeoutCb: null,

upgradeOrConnect: false,

parser: null,

maxHeadersCount: null,

reusedSocket: false,

host: 'localhost',

protocol: 'http:',

\_redirectable: [Circular \*2],

[Symbol(kCapture)]: false,

[Symbol(kBytesWritten)]: 0,

[Symbol(kEndCalled)]: true,

[Symbol(kNeedDrain)]: false,

[Symbol(corked)]: 0,

[Symbol(kOutHeaders)]: [Object: null prototype] {

accept: [ 'Accept', 'application/json, text/plain, \*/\*' ],

'content-type': [ 'Content-Type', 'application/x-www-form-urlencoded' ],

'user-agent': [ 'User-Agent', 'axios/1.3.3' ],

'accept-encoding': [ 'Accept-Encoding', 'gzip, compress, deflate, br' ],

host: [ 'Host', 'localhost:5000' ]

},

[Symbol(errored)]: null,

[Symbol(kUniqueHeaders)]: null

},

\_currentUrl: 'http://localhost:5000/user/generateToken',

[Symbol(kCapture)]: false

},

cause: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

}

}

Node.js v18.14.0

E:\nodejs>node axios4.js

E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825

AxiosError.call(axiosError, error.message, code, config, request, response);

^

AxiosError: connect ECONNREFUSED ::1:5000

at AxiosError.from (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825:14)

at RedirectableRequest.handleRequestError (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:2965:25)

at RedirectableRequest.emit (node:events:513:28)

at eventHandlers.<computed> (E:\nodejs\node\_modules\follow-redirects\index.js:14:24)

at ClientRequest.emit (node:events:513:28)

at Socket.socketErrorListener (node:\_http\_client:502:9)

at Socket.emit (node:events:513:28)

at emitErrorNT (node:internal/streams/destroy:151:8)

at emitErrorCloseNT (node:internal/streams/destroy:116:3)

at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) {

port: 5000,

address: '::1',

syscall: 'connect',

code: 'ECONNREFUSED',

errno: -4078,

config: {

transitional: {

silentJSONParsing: true,

forcedJSONParsing: true,

clarifyTimeoutError: false

},

adapter: [ 'xhr', 'http' ],

transformRequest: [ [Function: transformRequest] ],

transformResponse: [ [Function: transformResponse] ],

timeout: 0,

xsrfCookieName: 'XSRF-TOKEN',

xsrfHeaderName: 'X-XSRF-TOKEN',

maxContentLength: -1,

maxBodyLength: -1,

env: {

FormData: [Function: FormData] {

LINE\_BREAK: '\r\n',

DEFAULT\_CONTENT\_TYPE: 'application/octet-stream'

},

Blob: [class Blob]

},

validateStatus: [Function: validateStatus],

headers: AxiosHeaders {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

method: 'post',

url: 'http://localhost:5000/user/generateToken',

data: undefined

},

request: <ref \*2> Writable {

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: false,

decodeStrings: true,

defaultEncoding: 'utf8',

length: 0,

writing: false,

corked: 0,

sync: true,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: null,

writelen: 0,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 0,

constructed: true,

prefinished: false,

errorEmitted: false,

emitClose: true,

autoDestroy: true,

errored: null,

closed: false,

closeEmitted: false,

[Symbol(kOnFinished)]: []

},

\_events: [Object: null prototype] {

response: [Function: handleResponse],

error: [Function: handleRequestError],

socket: [Function: handleRequestSocket]

},

\_eventsCount: 3,

\_maxListeners: undefined,

\_options: {

maxRedirects: 21,

maxBodyLength: Infinity,

protocol: 'http:',

path: '/user/generateToken',

method: 'POST',

headers: [Object: null prototype] {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

agents: { http: undefined, https: undefined },

auth: undefined,

beforeRedirect: [Function: dispatchBeforeRedirect],

beforeRedirects: { proxy: [Function: beforeRedirect] },

hostname: 'localhost',

port: '5000',

agent: undefined,

nativeProtocols: {

'http:': {

\_connectionListener: [Function: connectionListener],

METHODS: [

'ACL', 'BIND', 'CHECKOUT',

'CONNECT', 'COPY', 'DELETE',

'GET', 'HEAD', 'LINK',

'LOCK', 'M-SEARCH', 'MERGE',

'MKACTIVITY', 'MKCALENDAR', 'MKCOL',

'MOVE', 'NOTIFY', 'OPTIONS',

'PATCH', 'POST', 'PROPFIND',

'PROPPATCH', 'PURGE', 'PUT',

'REBIND', 'REPORT', 'SEARCH',

'SOURCE', 'SUBSCRIBE', 'TRACE',

'UNBIND', 'UNLINK', 'UNLOCK',

'UNSUBSCRIBE'

],

STATUS\_CODES: {

'100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'103': 'Early Hints',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'208': 'Already Reported',

'226': 'IM Used',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Found',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'308': 'Permanent Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Timeout',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Payload Too Large',

'414': 'URI Too Long',

'415': 'Unsupported Media Type',

'416': 'Range Not Satisfiable',

'417': 'Expectation Failed',

'418': "I'm a Teapot",

'421': 'Misdirected Request',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Too Early',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'451': 'Unavailable For Legal Reasons',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Timeout',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'508': 'Loop Detected',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required'

},

Agent: [Function: Agent] { defaultMaxSockets: Infinity },

ClientRequest: [Function: ClientRequest],

IncomingMessage: [Function: IncomingMessage],

OutgoingMessage: [Function: OutgoingMessage],

Server: [Function: Server],

ServerResponse: [Function: ServerResponse],

createServer: [Function: createServer],

validateHeaderName: [Function: \_\_node\_internal\_],

validateHeaderValue: [Function: \_\_node\_internal\_],

get: [Function: get],

request: [Function: request],

setMaxIdleHTTPParsers: [Function: setMaxIdleHTTPParsers],

maxHeaderSize: [Getter],

globalAgent: [Getter/Setter]

},

'https:': {

Agent: [Function: Agent],

globalAgent: Agent {

\_events: [Object: null prototype],

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 443,

protocol: 'https:',

options: [Object: null prototype],

requests: [Object: null prototype] {},

sockets: [Object: null prototype] {},

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 0,

maxCachedSessions: 100,

\_sessionCache: [Object],

[Symbol(kCapture)]: false

},

Server: [Function: Server],

createServer: [Function: createServer],

get: [Function: get],

request: [Function: request]

}

},

pathname: '/user/generateToken'

},

\_ended: true,

\_ending: true,

\_redirectCount: 0,

\_redirects: [],

\_requestBodyLength: 0,

\_requestBodyBuffers: [],

\_onNativeResponse: [Function (anonymous)],

\_currentRequest: <ref \*1> ClientRequest {

\_events: [Object: null prototype] {

response: [Function: bound onceWrapper] {

listener: [Function (anonymous)]

},

abort: [Function (anonymous)],

aborted: [Function (anonymous)],

connect: [Function (anonymous)],

error: [Function (anonymous)],

socket: [Function (anonymous)],

timeout: [Function (anonymous)]

},

\_eventsCount: 7,

\_maxListeners: undefined,

outputData: [],

outputSize: 0,

writable: true,

destroyed: false,

\_last: true,

chunkedEncoding: false,

shouldKeepAlive: false,

maxRequestsOnConnectionReached: false,

\_defaultKeepAlive: true,

useChunkedEncodingByDefault: true,

sendDate: false,

\_removedConnection: false,

\_removedContLen: false,

\_removedTE: false,

strictContentLength: false,

\_contentLength: 0,

\_hasBody: true,

\_trailer: '',

finished: true,

\_headerSent: true,

\_closed: false,

socket: Socket {

connecting: false,

\_hadError: true,

\_parent: null,

\_host: 'localhost',

\_closeAfterHandlingError: false,

\_readableState: ReadableState {

objectMode: false,

highWaterMark: 16384,

buffer: BufferList { head: null, tail: null, length: 0 },

length: 0,

pipes: [],

flowing: true,

ended: false,

endEmitted: false,

reading: true,

constructed: true,

sync: false,

needReadable: true,

emittedReadable: false,

readableListening: false,

resumeScheduled: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

destroyed: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

defaultEncoding: 'utf8',

awaitDrainWriters: null,

multiAwaitDrain: false,

readingMore: false,

dataEmitted: false,

decoder: null,

encoding: null,

[Symbol(kPaused)]: false

},

\_events: [Object: null prototype] {

end: [Function: onReadableStreamEnd],

connect: [ [Function], [Function], [Function] ],

free: [Function: onFree],

close: [

[Function: onClose],

[Function: socketCloseListener],

[Function]

],

timeout: [Function: onTimeout],

agentRemove: [Function: onRemove],

error: [Function: socketErrorListener],

drain: [Function: ondrain]

},

\_eventsCount: 8,

\_maxListeners: undefined,

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: true,

decodeStrings: false,

defaultEncoding: 'utf8',

length: 260,

writing: true,

corked: 0,

sync: false,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: [Function: bound onFinish],

writelen: 260,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 1,

constructed: true,

prefinished: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

[Symbol(kOnFinished)]: []

},

allowHalfOpen: false,

\_sockname: null,

\_pendingData: 'POST /user/generateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_pendingEncoding: 'latin1',

server: null,

\_server: null,

parser: null,

\_httpMessage: [Circular \*1],

[Symbol(async\_id\_symbol)]: 4,

[Symbol(kHandle)]: null,

[Symbol(lastWriteQueueSize)]: 0,

[Symbol(timeout)]: null,

[Symbol(kBuffer)]: null,

[Symbol(kBufferCb)]: null,

[Symbol(kBufferGen)]: null,

[Symbol(kCapture)]: false,

[Symbol(kSetNoDelay)]: true,

[Symbol(kSetKeepAlive)]: true,

[Symbol(kSetKeepAliveInitialDelay)]: 60,

[Symbol(kBytesRead)]: 0,

[Symbol(kBytesWritten)]: 0

},

\_header: 'POST /user/generateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_keepAliveTimeout: 0,

\_onPendingData: [Function: nop],

agent: Agent {

\_events: [Object: null prototype] {

free: [Function (anonymous)],

newListener: [Function: maybeEnableKeylog]

},

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 80,

protocol: 'http:',

options: [Object: null prototype] { noDelay: true, path: null },

requests: [Object: null prototype] {},

sockets: [Object: null prototype] { 'localhost:5000:': [ [Socket] ] },

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 1,

[Symbol(kCapture)]: false

},

socketPath: undefined,

method: 'POST',

maxHeaderSize: undefined,

insecureHTTPParser: undefined,

joinDuplicateHeaders: undefined,

path: '/user/generateToken',

\_ended: false,

res: null,

aborted: false,

timeoutCb: null,

upgradeOrConnect: false,

parser: null,

maxHeadersCount: null,

reusedSocket: false,

host: 'localhost',

protocol: 'http:',

\_redirectable: [Circular \*2],

[Symbol(kCapture)]: false,

[Symbol(kBytesWritten)]: 0,

[Symbol(kEndCalled)]: true,

[Symbol(kNeedDrain)]: false,

[Symbol(corked)]: 0,

[Symbol(kOutHeaders)]: [Object: null prototype] {

accept: [ 'Accept', 'application/json, text/plain, \*/\*' ],

'content-type': [ 'Content-Type', 'application/x-www-form-urlencoded' ],

'user-agent': [ 'User-Agent', 'axios/1.3.3' ],

'accept-encoding': [ 'Accept-Encoding', 'gzip, compress, deflate, br' ],

host: [ 'Host', 'localhost:5000' ]

},

[Symbol(errored)]: null,

[Symbol(kUniqueHeaders)]: null

},

\_currentUrl: 'http://localhost:5000/user/generateToken',

[Symbol(kCapture)]: false

},

cause: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

}

}

Node.js v18.14.0

E:\nodejs>

E:\nodejs>node index4.js

Server is up and running on 5000 ...

^C

E:\nodejs>node axios4.js

E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825

AxiosError.call(axiosError, error.message, code, config, request, response);

^

AxiosError: connect ECONNREFUSED ::1:5000

at AxiosError.from (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825:14)

at RedirectableRequest.handleRequestError (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:2965:25)

at RedirectableRequest.emit (node:events:513:28)

at eventHandlers.<computed> (E:\nodejs\node\_modules\follow-redirects\index.js:14:24)

at ClientRequest.emit (node:events:513:28)

at Socket.socketErrorListener (node:\_http\_client:502:9)

at Socket.emit (node:events:513:28)

at emitErrorNT (node:internal/streams/destroy:151:8)

at emitErrorCloseNT (node:internal/streams/destroy:116:3)

at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) {

port: 5000,

address: '::1',

syscall: 'connect',

code: 'ECONNREFUSED',

errno: -4078,

config: {

transitional: {

silentJSONParsing: true,

forcedJSONParsing: true,

clarifyTimeoutError: false

},

adapter: [ 'xhr', 'http' ],

transformRequest: [ [Function: transformRequest] ],

transformResponse: [ [Function: transformResponse] ],

timeout: 0,

xsrfCookieName: 'XSRF-TOKEN',

xsrfHeaderName: 'X-XSRF-TOKEN',

maxContentLength: -1,

maxBodyLength: -1,

env: {

FormData: [Function: FormData] {

LINE\_BREAK: '\r\n',

DEFAULT\_CONTENT\_TYPE: 'application/octet-stream'

},

Blob: [class Blob]

},

validateStatus: [Function: validateStatus],

headers: AxiosHeaders {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

method: 'post',

url: 'http://localhost:5000/user/generateToken',

data: undefined

},

request: <ref \*2> Writable {

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: false,

decodeStrings: true,

defaultEncoding: 'utf8',

length: 0,

writing: false,

corked: 0,

sync: true,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: null,

writelen: 0,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 0,

constructed: true,

prefinished: false,

errorEmitted: false,

emitClose: true,

autoDestroy: true,

errored: null,

closed: false,

closeEmitted: false,

[Symbol(kOnFinished)]: []

},

\_events: [Object: null prototype] {

response: [Function: handleResponse],

error: [Function: handleRequestError],

socket: [Function: handleRequestSocket]

},

\_eventsCount: 3,

\_maxListeners: undefined,

\_options: {

maxRedirects: 21,

maxBodyLength: Infinity,

protocol: 'http:',

path: '/user/generateToken',

method: 'POST',

headers: [Object: null prototype] {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

agents: { http: undefined, https: undefined },

auth: undefined,

beforeRedirect: [Function: dispatchBeforeRedirect],

beforeRedirects: { proxy: [Function: beforeRedirect] },

hostname: 'localhost',

port: '5000',

agent: undefined,

nativeProtocols: {

'http:': {

\_connectionListener: [Function: connectionListener],

METHODS: [

'ACL', 'BIND', 'CHECKOUT',

'CONNECT', 'COPY', 'DELETE',

'GET', 'HEAD', 'LINK',

'LOCK', 'M-SEARCH', 'MERGE',

'MKACTIVITY', 'MKCALENDAR', 'MKCOL',

'MOVE', 'NOTIFY', 'OPTIONS',

'PATCH', 'POST', 'PROPFIND',

'PROPPATCH', 'PURGE', 'PUT',

'REBIND', 'REPORT', 'SEARCH',

'SOURCE', 'SUBSCRIBE', 'TRACE',

'UNBIND', 'UNLINK', 'UNLOCK',

'UNSUBSCRIBE'

],

STATUS\_CODES: {

'100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'103': 'Early Hints',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'208': 'Already Reported',

'226': 'IM Used',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Found',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'308': 'Permanent Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Timeout',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Payload Too Large',

'414': 'URI Too Long',

'415': 'Unsupported Media Type',

'416': 'Range Not Satisfiable',

'417': 'Expectation Failed',

'418': "I'm a Teapot",

'421': 'Misdirected Request',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Too Early',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'451': 'Unavailable For Legal Reasons',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Timeout',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'508': 'Loop Detected',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required'

},

Agent: [Function: Agent] { defaultMaxSockets: Infinity },

ClientRequest: [Function: ClientRequest],

IncomingMessage: [Function: IncomingMessage],

OutgoingMessage: [Function: OutgoingMessage],

Server: [Function: Server],

ServerResponse: [Function: ServerResponse],

createServer: [Function: createServer],

validateHeaderName: [Function: \_\_node\_internal\_],

validateHeaderValue: [Function: \_\_node\_internal\_],

get: [Function: get],

request: [Function: request],

setMaxIdleHTTPParsers: [Function: setMaxIdleHTTPParsers],

maxHeaderSize: [Getter],

globalAgent: [Getter/Setter]

},

'https:': {

Agent: [Function: Agent],

globalAgent: Agent {

\_events: [Object: null prototype],

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 443,

protocol: 'https:',

options: [Object: null prototype],

requests: [Object: null prototype] {},

sockets: [Object: null prototype] {},

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 0,

maxCachedSessions: 100,

\_sessionCache: [Object],

[Symbol(kCapture)]: false

},

Server: [Function: Server],

createServer: [Function: createServer],

get: [Function: get],

request: [Function: request]

}

},

pathname: '/user/generateToken'

},

\_ended: true,

\_ending: true,

\_redirectCount: 0,

\_redirects: [],

\_requestBodyLength: 0,

\_requestBodyBuffers: [],

\_onNativeResponse: [Function (anonymous)],

\_currentRequest: <ref \*1> ClientRequest {

\_events: [Object: null prototype] {

response: [Function: bound onceWrapper] {

listener: [Function (anonymous)]

},

abort: [Function (anonymous)],

aborted: [Function (anonymous)],

connect: [Function (anonymous)],

error: [Function (anonymous)],

socket: [Function (anonymous)],

timeout: [Function (anonymous)]

},

\_eventsCount: 7,

\_maxListeners: undefined,

outputData: [],

outputSize: 0,

writable: true,

destroyed: false,

\_last: true,

chunkedEncoding: false,

shouldKeepAlive: false,

maxRequestsOnConnectionReached: false,

\_defaultKeepAlive: true,

useChunkedEncodingByDefault: true,

sendDate: false,

\_removedConnection: false,

\_removedContLen: false,

\_removedTE: false,

strictContentLength: false,

\_contentLength: 0,

\_hasBody: true,

\_trailer: '',

finished: true,

\_headerSent: true,

\_closed: false,

socket: Socket {

connecting: false,

\_hadError: true,

\_parent: null,

\_host: 'localhost',

\_closeAfterHandlingError: false,

\_readableState: ReadableState {

objectMode: false,

highWaterMark: 16384,

buffer: BufferList { head: null, tail: null, length: 0 },

length: 0,

pipes: [],

flowing: true,

ended: false,

endEmitted: false,

reading: true,

constructed: true,

sync: false,

needReadable: true,

emittedReadable: false,

readableListening: false,

resumeScheduled: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

destroyed: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

defaultEncoding: 'utf8',

awaitDrainWriters: null,

multiAwaitDrain: false,

readingMore: false,

dataEmitted: false,

decoder: null,

encoding: null,

[Symbol(kPaused)]: false

},

\_events: [Object: null prototype] {

end: [Function: onReadableStreamEnd],

connect: [ [Function], [Function], [Function] ],

free: [Function: onFree],

close: [

[Function: onClose],

[Function: socketCloseListener],

[Function]

],

timeout: [Function: onTimeout],

agentRemove: [Function: onRemove],

error: [Function: socketErrorListener],

drain: [Function: ondrain]

},

\_eventsCount: 8,

\_maxListeners: undefined,

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: true,

decodeStrings: false,

defaultEncoding: 'utf8',

length: 260,

writing: true,

corked: 0,

sync: false,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: [Function: bound onFinish],

writelen: 260,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 1,

constructed: true,

prefinished: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

[Symbol(kOnFinished)]: []

},

allowHalfOpen: false,

\_sockname: null,

\_pendingData: 'POST /user/generateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_pendingEncoding: 'latin1',

server: null,

\_server: null,

parser: null,

\_httpMessage: [Circular \*1],

[Symbol(async\_id\_symbol)]: 4,

[Symbol(kHandle)]: null,

[Symbol(lastWriteQueueSize)]: 0,

[Symbol(timeout)]: null,

[Symbol(kBuffer)]: null,

[Symbol(kBufferCb)]: null,

[Symbol(kBufferGen)]: null,

[Symbol(kCapture)]: false,

[Symbol(kSetNoDelay)]: true,

[Symbol(kSetKeepAlive)]: true,

[Symbol(kSetKeepAliveInitialDelay)]: 60,

[Symbol(kBytesRead)]: 0,

[Symbol(kBytesWritten)]: 0

},

\_header: 'POST /user/generateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_keepAliveTimeout: 0,

\_onPendingData: [Function: nop],

agent: Agent {

\_events: [Object: null prototype] {

free: [Function (anonymous)],

newListener: [Function: maybeEnableKeylog]

},

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 80,

protocol: 'http:',

options: [Object: null prototype] { noDelay: true, path: null },

requests: [Object: null prototype] {},

sockets: [Object: null prototype] { 'localhost:5000:': [ [Socket] ] },

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 1,

[Symbol(kCapture)]: false

},

socketPath: undefined,

method: 'POST',

maxHeaderSize: undefined,

insecureHTTPParser: undefined,

joinDuplicateHeaders: undefined,

path: '/user/generateToken',

\_ended: false,

res: null,

aborted: false,

timeoutCb: null,

upgradeOrConnect: false,

parser: null,

maxHeadersCount: null,

reusedSocket: false,

host: 'localhost',

protocol: 'http:',

\_redirectable: [Circular \*2],

[Symbol(kCapture)]: false,

[Symbol(kBytesWritten)]: 0,

[Symbol(kEndCalled)]: true,

[Symbol(kNeedDrain)]: false,

[Symbol(corked)]: 0,

[Symbol(kOutHeaders)]: [Object: null prototype] {

accept: [ 'Accept', 'application/json, text/plain, \*/\*' ],

'content-type': [ 'Content-Type', 'application/x-www-form-urlencoded' ],

'user-agent': [ 'User-Agent', 'axios/1.3.3' ],

'accept-encoding': [ 'Accept-Encoding', 'gzip, compress, deflate, br' ],

host: [ 'Host', 'localhost:5000' ]

},

[Symbol(errored)]: null,

[Symbol(kUniqueHeaders)]: null

},

\_currentUrl: 'http://localhost:5000/user/generateToken',

[Symbol(kCapture)]: false

},

cause: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

}

}

Node.js v18.14.0

E:\nodejs>node axios5.js

E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825

AxiosError.call(axiosError, error.message, code, config, request, response);

^

AxiosError: connect ECONNREFUSED ::1:5000

at AxiosError.from (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825:14)

at RedirectableRequest.handleRequestError (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:2965:25)

at RedirectableRequest.emit (node:events:513:28)

at eventHandlers.<computed> (E:\nodejs\node\_modules\follow-redirects\index.js:14:24)

at ClientRequest.emit (node:events:513:28)

at Socket.socketErrorListener (node:\_http\_client:502:9)

at Socket.emit (node:events:513:28)

at emitErrorNT (node:internal/streams/destroy:151:8)

at emitErrorCloseNT (node:internal/streams/destroy:116:3)

at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) {

port: 5000,

address: '::1',

syscall: 'connect',

code: 'ECONNREFUSED',

errno: -4078,

config: {

transitional: {

silentJSONParsing: true,

forcedJSONParsing: true,

clarifyTimeoutError: false

},

adapter: [ 'xhr', 'http' ],

transformRequest: [ [Function: transformRequest] ],

transformResponse: [ [Function: transformResponse] ],

timeout: 0,

xsrfCookieName: 'XSRF-TOKEN',

xsrfHeaderName: 'X-XSRF-TOKEN',

maxContentLength: -1,

maxBodyLength: -1,

env: {

FormData: [Function: FormData] {

LINE\_BREAK: '\r\n',

DEFAULT\_CONTENT\_TYPE: 'application/octet-stream'

},

Blob: [class Blob]

},

validateStatus: [Function: validateStatus],

headers: AxiosHeaders {

Accept: 'application/json, text/plain, \*/\*',

Authorization: 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

method: 'get',

url: 'http://localhost:5000/user/validateToken',

data: undefined

},

request: <ref \*2> Writable {

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: false,

decodeStrings: true,

defaultEncoding: 'utf8',

length: 0,

writing: false,

corked: 0,

sync: true,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: null,

writelen: 0,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 0,

constructed: true,

prefinished: false,

errorEmitted: false,

emitClose: true,

autoDestroy: true,

errored: null,

closed: false,

closeEmitted: false,

[Symbol(kOnFinished)]: []

},

\_events: [Object: null prototype] {

response: [Function: handleResponse],

error: [Function: handleRequestError],

socket: [Function: handleRequestSocket]

},

\_eventsCount: 3,

\_maxListeners: undefined,

\_options: {

maxRedirects: 21,

maxBodyLength: Infinity,

protocol: 'http:',

path: '/user/validateToken',

method: 'GET',

headers: [Object: null prototype] {

Accept: 'application/json, text/plain, \*/\*',

Authorization: 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

agents: { http: undefined, https: undefined },

auth: undefined,

beforeRedirect: [Function: dispatchBeforeRedirect],

beforeRedirects: { proxy: [Function: beforeRedirect] },

hostname: 'localhost',

port: '5000',

agent: undefined,

nativeProtocols: {

'http:': {

\_connectionListener: [Function: connectionListener],

METHODS: [

'ACL', 'BIND', 'CHECKOUT',

'CONNECT', 'COPY', 'DELETE',

'GET', 'HEAD', 'LINK',

'LOCK', 'M-SEARCH', 'MERGE',

'MKACTIVITY', 'MKCALENDAR', 'MKCOL',

'MOVE', 'NOTIFY', 'OPTIONS',

'PATCH', 'POST', 'PROPFIND',

'PROPPATCH', 'PURGE', 'PUT',

'REBIND', 'REPORT', 'SEARCH',

'SOURCE', 'SUBSCRIBE', 'TRACE',

'UNBIND', 'UNLINK', 'UNLOCK',

'UNSUBSCRIBE'

],

STATUS\_CODES: {

'100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'103': 'Early Hints',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'208': 'Already Reported',

'226': 'IM Used',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Found',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'308': 'Permanent Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Timeout',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Payload Too Large',

'414': 'URI Too Long',

'415': 'Unsupported Media Type',

'416': 'Range Not Satisfiable',

'417': 'Expectation Failed',

'418': "I'm a Teapot",

'421': 'Misdirected Request',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Too Early',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'451': 'Unavailable For Legal Reasons',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Timeout',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'508': 'Loop Detected',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required'

},

Agent: [Function: Agent] { defaultMaxSockets: Infinity },

ClientRequest: [Function: ClientRequest],

IncomingMessage: [Function: IncomingMessage],

OutgoingMessage: [Function: OutgoingMessage],

Server: [Function: Server],

ServerResponse: [Function: ServerResponse],

createServer: [Function: createServer],

validateHeaderName: [Function: \_\_node\_internal\_],

validateHeaderValue: [Function: \_\_node\_internal\_],

get: [Function: get],

request: [Function: request],

setMaxIdleHTTPParsers: [Function: setMaxIdleHTTPParsers],

maxHeaderSize: [Getter],

globalAgent: [Getter/Setter]

},

'https:': {

Agent: [Function: Agent],

globalAgent: Agent {

\_events: [Object: null prototype],

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 443,

protocol: 'https:',

options: [Object: null prototype],

requests: [Object: null prototype] {},

sockets: [Object: null prototype] {},

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 0,

maxCachedSessions: 100,

\_sessionCache: [Object],

[Symbol(kCapture)]: false

},

Server: [Function: Server],

createServer: [Function: createServer],

get: [Function: get],

request: [Function: request]

}

},

pathname: '/user/validateToken'

},

\_ended: true,

\_ending: true,

\_redirectCount: 0,

\_redirects: [],

\_requestBodyLength: 0,

\_requestBodyBuffers: [],

\_onNativeResponse: [Function (anonymous)],

\_currentRequest: <ref \*1> ClientRequest {

\_events: [Object: null prototype] {

response: [Function: bound onceWrapper] {

listener: [Function (anonymous)]

},

abort: [Function (anonymous)],

aborted: [Function (anonymous)],

connect: [Function (anonymous)],

error: [Function (anonymous)],

socket: [Function (anonymous)],

timeout: [Function (anonymous)]

},

\_eventsCount: 7,

\_maxListeners: undefined,

outputData: [],

outputSize: 0,

writable: true,

destroyed: false,

\_last: true,

chunkedEncoding: false,

shouldKeepAlive: false,

maxRequestsOnConnectionReached: false,

\_defaultKeepAlive: true,

useChunkedEncodingByDefault: false,

sendDate: false,

\_removedConnection: false,

\_removedContLen: false,

\_removedTE: false,

strictContentLength: false,

\_contentLength: 0,

\_hasBody: true,

\_trailer: '',

finished: true,

\_headerSent: true,

\_closed: false,

socket: Socket {

connecting: false,

\_hadError: true,

\_parent: null,

\_host: 'localhost',

\_closeAfterHandlingError: false,

\_readableState: ReadableState {

objectMode: false,

highWaterMark: 16384,

buffer: BufferList { head: null, tail: null, length: 0 },

length: 0,

pipes: [],

flowing: true,

ended: false,

endEmitted: false,

reading: true,

constructed: true,

sync: false,

needReadable: true,

emittedReadable: false,

readableListening: false,

resumeScheduled: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

destroyed: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

defaultEncoding: 'utf8',

awaitDrainWriters: null,

multiAwaitDrain: false,

readingMore: false,

dataEmitted: false,

decoder: null,

encoding: null,

[Symbol(kPaused)]: false

},

\_events: [Object: null prototype] {

end: [Function: onReadableStreamEnd],

connect: [ [Function], [Function], [Function] ],

free: [Function: onFree],

close: [

[Function: onClose],

[Function: socketCloseListener],

[Function]

],

timeout: [Function: onTimeout],

agentRemove: [Function: onRemove],

error: [Function: socketErrorListener],

drain: [Function: ondrain]

},

\_eventsCount: 8,

\_maxListeners: undefined,

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: true,

decodeStrings: false,

defaultEncoding: 'utf8',

length: 423,

writing: true,

corked: 0,

sync: false,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: [Function: bound onFinish],

writelen: 423,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 1,

constructed: true,

prefinished: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

[Symbol(kOnFinished)]: []

},

allowHalfOpen: false,

\_sockname: null,

\_pendingData: 'GET /user/validateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'\r\n',

\_pendingEncoding: 'latin1',

server: null,

\_server: null,

parser: null,

\_httpMessage: [Circular \*1],

[Symbol(async\_id\_symbol)]: 4,

[Symbol(kHandle)]: null,

[Symbol(lastWriteQueueSize)]: 0,

[Symbol(timeout)]: null,

[Symbol(kBuffer)]: null,

[Symbol(kBufferCb)]: null,

[Symbol(kBufferGen)]: null,

[Symbol(kCapture)]: false,

[Symbol(kSetNoDelay)]: true,

[Symbol(kSetKeepAlive)]: true,

[Symbol(kSetKeepAliveInitialDelay)]: 60,

[Symbol(kBytesRead)]: 0,

[Symbol(kBytesWritten)]: 0

},

\_header: 'GET /user/validateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'\r\n',

\_keepAliveTimeout: 0,

\_onPendingData: [Function: nop],

agent: Agent {

\_events: [Object: null prototype] {

free: [Function (anonymous)],

newListener: [Function: maybeEnableKeylog]

},

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 80,

protocol: 'http:',

options: [Object: null prototype] { noDelay: true, path: null },

requests: [Object: null prototype] {},

sockets: [Object: null prototype] { 'localhost:5000:': [ [Socket] ] },

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 1,

[Symbol(kCapture)]: false

},

socketPath: undefined,

method: 'GET',

maxHeaderSize: undefined,

insecureHTTPParser: undefined,

joinDuplicateHeaders: undefined,

path: '/user/validateToken',

\_ended: false,

res: null,

aborted: false,

timeoutCb: null,

upgradeOrConnect: false,

parser: null,

maxHeadersCount: null,

reusedSocket: false,

host: 'localhost',

protocol: 'http:',

\_redirectable: [Circular \*2],

[Symbol(kCapture)]: false,

[Symbol(kBytesWritten)]: 0,

[Symbol(kEndCalled)]: true,

[Symbol(kNeedDrain)]: false,

[Symbol(corked)]: 0,

[Symbol(kOutHeaders)]: [Object: null prototype] {

accept: [ 'Accept', 'application/json, text/plain, \*/\*' ],

authorization: [

'Authorization',

'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA'

],

'user-agent': [ 'User-Agent', 'axios/1.3.3' ],

'accept-encoding': [ 'Accept-Encoding', 'gzip, compress, deflate, br' ],

host: [ 'Host', 'localhost:5000' ]

},

[Symbol(errored)]: null,

[Symbol(kUniqueHeaders)]: null

},

\_currentUrl: 'http://localhost:5000/user/validateToken',

[Symbol(kCapture)]: false

},

cause: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

}

}

Node.js v18.14.0

E:\nodejs>curl -H "Content-type: multipart/form-data" -X POST http://localhost:5000/user/generateToken

curl: (7) Failed to connect to localhost port 5000 after 2273 ms: Connection refused

E:\nodejs>dir

Volume in drive E has no label.

Volume Serial Number is D440-B9AA

Directory of E:\nodejs

02/17/2023 11:30 AM <DIR> .

02/17/2023 11:30 AM <DIR> ..

02/15/2023 02:47 PM 73 app.js

02/15/2023 03:21 PM 81 app.ts

02/15/2023 12:40 PM 161 app2.vue

02/17/2023 10:21 AM 122 axios1.js

02/17/2023 10:30 AM 116 axios2.js

02/17/2023 11:17 AM 138 axios3.js

02/17/2023 11:27 AM 138 axios4.js

02/17/2023 11:30 AM 398 axios5.js

02/15/2023 10:40 AM 5,672 bird150.jpg

02/15/2023 10:17 AM 1,786 bootclounm.html

02/15/2023 11:05 AM 1,779 bootclounmimage.html

02/15/2023 10:13 AM 1,391 bootstrap.html

02/15/2023 10:45 AM 1,556 boottheem2.html

02/15/2023 10:44 AM 669 boottheme.html

02/15/2023 10:39 AM 1,788 carousel.html

02/15/2023 10:37 AM 71,294 chicago.jpg

02/15/2023 10:16 AM 89,682 cinqueterre.jpg

02/17/2023 09:59 AM 652 example1.js

02/16/2023 03:48 PM 520 example2.js

02/16/2023 03:26 PM 302 index.html

02/16/2023 03:38 PM 784 index1.html

02/16/2023 04:22 PM 1,221 index3.html

02/17/2023 10:51 AM 1,241 index4.js

02/17/2023 11:21 AM 138 index5.html

02/15/2023 10:36 AM 22,914 la.jpg

02/15/2023 11:39 AM 999 leader.html

02/14/2023 03:21 PM 185 main1.js

02/14/2023 03:11 PM 56 module1.js

02/14/2023 03:13 PM 549 module2.js

02/14/2023 03:21 PM 14 mynewfile.txt

02/15/2023 10:29 AM 33,636 newyork.jpg

02/14/2023 12:51 PM 175 node1.js

02/14/2023 01:00 PM 440 node11.js

02/14/2023 02:42 PM 457 node2.js

02/14/2023 03:23 PM 643 node3.js

02/17/2023 10:47 AM <DIR> node\_modules

02/15/2023 10:37 AM 53,292 ny.jpg

02/17/2023 10:47 AM 45,301 package-lock.json

02/17/2023 10:47 AM 533 package.json

02/16/2023 04:21 PM 260 package1.json

02/16/2023 04:13 PM 1,066 routing\_example.js

02/15/2023 10:29 AM 16,102 sanfran.jpg

02/16/2023 04:24 PM 849 server.js

02/15/2023 11:12 AM 5,415 sw.jpeg

02/15/2023 11:39 AM 766 threeimgs.html

02/15/2023 02:30 PM 11,401 tsconfig.json

02/15/2023 11:54 AM <DIR> y

45 File(s) 376,755 bytes

4 Dir(s) 471,839,375,360 bytes free

E:\nodejs>cd:

The filename, directory name, or volume label syntax is incorrect.

E:\nodejs>cd:

The filename, directory name, or volume label syntax is incorrect.

E:\nodejs>c:

C:\>dir

Volume in drive C has no label.

Volume Serial Number is 78FE-8B58

Directory of C:\

04/17/2022 07:34 PM <DIR> Autodesk

01/06/2021 05:05 PM <DIR> Drivers

02/16/2023 04:30 PM <DIR> Jsonnew

12/07/2019 02:44 PM <DIR> PerfLogs

02/16/2023 11:30 AM <DIR> Program Files

02/14/2023 03:54 PM <DIR> Program Files (x86)

07/15/2021 12:26 AM <DIR> SProV8i SS6

07/15/2021 12:59 AM <DIR> Staad.foundation 5.3

02/17/2023 10:27 AM <DIR> Tamil

04/25/2021 03:17 PM <DIR> Users

02/14/2023 04:12 PM <DIR> Windows

04/18/2021 08:31 PM <DIR> Windows10Upgrade

08/26/2022 06:58 PM <DIR> xampp

0 File(s) 0 bytes

13 Dir(s) 383,498,285,056 bytes free

C:\>curl -H "Content-type: multipart/form-data" -X POST http://localhost:5000/user/generateToken

curl: (7) Failed to connect to localhost port 5000 after 2254 ms: Connection refused

C:\>E:\

'E:\' is not recognized as an internal or external command,

operable program or batch file.

C:\>cd E:

E:\nodejs

C:\>E:

E:\nodejs>curl -v

curl: no URL specified!

curl: try 'curl --help' for more information

E:\nodejs>curl --help

Usage: curl [options...] <url>

-d, --data <data> HTTP POST data

-f, --fail Fail fast with no output on HTTP errors

-h, --help <category> Get help for commands

-i, --include Include protocol response headers in the output

-o, --output <file> Write to file instead of stdout

-O, --remote-name Write output to a file named as the remote file

-s, --silent Silent mode

-T, --upload-file <file> Transfer local FILE to destination

-u, --user <user:password> Server user and password

-A, --user-agent <name> Send User-Agent <name> to server

-v, --verbose Make the operation more talkative

-V, --version Show version number and quit

This is not the full help, this menu is stripped into categories.

Use "--help category" to get an overview of all categories.

For all options use the manual or "--help all".

E:\nodejs>curl -h

Usage: curl [options...] <url>

-d, --data <data> HTTP POST data

-f, --fail Fail fast with no output on HTTP errors

-h, --help <category> Get help for commands

-i, --include Include protocol response headers in the output

-o, --output <file> Write to file instead of stdout

-O, --remote-name Write output to a file named as the remote file

-s, --silent Silent mode

-T, --upload-file <file> Transfer local FILE to destination

-u, --user <user:password> Server user and password

-A, --user-agent <name> Send User-Agent <name> to server

-v, --verbose Make the operation more talkative

-V, --version Show version number and quit

This is not the full help, this menu is stripped into categories.

Use "--help category" to get an overview of all categories.

For all options use the manual or "--help all".

E:\nodejs>curl --v

curl: option --v: is ambiguous

curl: try 'curl --help' for more information

E:\nodejs>curl --V

curl: option --V: is ambiguous

curl: try 'curl --help' for more information

E:\nodejs>curl -V

curl 7.83.1 (Windows) libcurl/7.83.1 Schannel

Release-Date: 2022-05-13

Protocols: dict file ftp ftps http https imap imaps pop3 pop3s smtp smtps telnet tftp

Features: AsynchDNS HSTS IPv6 Kerberos Largefile NTLM SPNEGO SSL SSPI UnixSockets

E:\nodejs>curl

curl: try 'curl --help' for more information

E:\nodejs>dir

Volume in drive E has no label.

Volume Serial Number is D440-B9AA

Directory of E:\nodejs

02/17/2023 11:30 AM <DIR> .

02/17/2023 11:30 AM <DIR> ..

02/15/2023 02:47 PM 73 app.js

02/15/2023 03:21 PM 81 app.ts

02/15/2023 12:40 PM 161 app2.vue

02/17/2023 10:21 AM 122 axios1.js

02/17/2023 10:30 AM 116 axios2.js

02/17/2023 11:17 AM 138 axios3.js

02/17/2023 11:27 AM 138 axios4.js

02/17/2023 11:30 AM 398 axios5.js

02/15/2023 10:40 AM 5,672 bird150.jpg

02/15/2023 10:17 AM 1,786 bootclounm.html

02/15/2023 11:05 AM 1,779 bootclounmimage.html

02/15/2023 10:13 AM 1,391 bootstrap.html

02/15/2023 10:45 AM 1,556 boottheem2.html

02/15/2023 10:44 AM 669 boottheme.html

02/15/2023 10:39 AM 1,788 carousel.html

02/15/2023 10:37 AM 71,294 chicago.jpg

02/15/2023 10:16 AM 89,682 cinqueterre.jpg

02/17/2023 09:59 AM 652 example1.js

02/16/2023 03:48 PM 520 example2.js

02/16/2023 03:26 PM 302 index.html

02/16/2023 03:38 PM 784 index1.html

02/16/2023 04:22 PM 1,221 index3.html

02/17/2023 10:51 AM 1,241 index4.js

02/17/2023 11:21 AM 138 index5.html

02/15/2023 10:36 AM 22,914 la.jpg

02/15/2023 11:39 AM 999 leader.html

02/14/2023 03:21 PM 185 main1.js

02/14/2023 03:11 PM 56 module1.js

02/14/2023 03:13 PM 549 module2.js

02/14/2023 03:21 PM 14 mynewfile.txt

02/15/2023 10:29 AM 33,636 newyork.jpg

02/14/2023 12:51 PM 175 node1.js

02/14/2023 01:00 PM 440 node11.js

02/14/2023 02:42 PM 457 node2.js

02/14/2023 03:23 PM 643 node3.js

02/17/2023 10:47 AM <DIR> node\_modules

02/15/2023 10:37 AM 53,292 ny.jpg

02/17/2023 10:47 AM 45,301 package-lock.json

02/17/2023 10:47 AM 533 package.json

02/16/2023 04:21 PM 260 package1.json

02/16/2023 04:13 PM 1,066 routing\_example.js

02/15/2023 10:29 AM 16,102 sanfran.jpg

02/16/2023 04:24 PM 849 server.js

02/15/2023 11:12 AM 5,415 sw.jpeg

02/15/2023 11:39 AM 766 threeimgs.html

02/15/2023 02:30 PM 11,401 tsconfig.json

02/15/2023 11:54 AM <DIR> y

45 File(s) 376,755 bytes

4 Dir(s) 471,839,375,360 bytes free

E:\nodejs>

E:\nodejs>

E:\nodejs>type .env

The system cannot find the file specified.

E:\nodejs>dir

Volume in drive E has no label.

Volume Serial Number is D440-B9AA

Directory of E:\nodejs

02/17/2023 11:30 AM <DIR> .

02/17/2023 11:30 AM <DIR> ..

02/15/2023 02:47 PM 73 app.js

02/15/2023 03:21 PM 81 app.ts

02/15/2023 12:40 PM 161 app2.vue

02/17/2023 10:21 AM 122 axios1.js

02/17/2023 10:30 AM 116 axios2.js

02/17/2023 11:17 AM 138 axios3.js

02/17/2023 11:27 AM 138 axios4.js

02/17/2023 11:30 AM 398 axios5.js

02/15/2023 10:40 AM 5,672 bird150.jpg

02/15/2023 10:17 AM 1,786 bootclounm.html

02/15/2023 11:05 AM 1,779 bootclounmimage.html

02/15/2023 10:13 AM 1,391 bootstrap.html

02/15/2023 10:45 AM 1,556 boottheem2.html

02/15/2023 10:44 AM 669 boottheme.html

02/15/2023 10:39 AM 1,788 carousel.html

02/15/2023 10:37 AM 71,294 chicago.jpg

02/15/2023 10:16 AM 89,682 cinqueterre.jpg

02/17/2023 09:59 AM 652 example1.js

02/16/2023 03:48 PM 520 example2.js

02/16/2023 03:26 PM 302 index.html

02/16/2023 03:38 PM 784 index1.html

02/16/2023 04:22 PM 1,221 index3.html

02/17/2023 10:51 AM 1,241 index4.js

02/17/2023 11:21 AM 138 index5.html

02/15/2023 10:36 AM 22,914 la.jpg

02/15/2023 11:39 AM 999 leader.html

02/14/2023 03:21 PM 185 main1.js

02/14/2023 03:11 PM 56 module1.js

02/14/2023 03:13 PM 549 module2.js

02/14/2023 03:21 PM 14 mynewfile.txt

02/15/2023 10:29 AM 33,636 newyork.jpg

02/14/2023 12:51 PM 175 node1.js

02/14/2023 01:00 PM 440 node11.js

02/14/2023 02:42 PM 457 node2.js

02/14/2023 03:23 PM 643 node3.js

02/17/2023 10:47 AM <DIR> node\_modules

02/15/2023 10:37 AM 53,292 ny.jpg

02/17/2023 10:47 AM 45,301 package-lock.json

02/17/2023 10:47 AM 533 package.json

02/16/2023 04:21 PM 260 package1.json

02/16/2023 04:13 PM 1,066 routing\_example.js

02/15/2023 10:29 AM 16,102 sanfran.jpg

02/16/2023 04:24 PM 849 server.js

02/15/2023 11:12 AM 5,415 sw.jpeg

02/15/2023 11:39 AM 766 threeimgs.html

02/15/2023 02:30 PM 11,401 tsconfig.json

02/15/2023 11:54 AM <DIR> y

45 File(s) 376,755 bytes

4 Dir(s) 471,839,375,360 bytes free

E:\nodejs>C:

C:\>dir

Volume in drive C has no label.

Volume Serial Number is 78FE-8B58

Directory of C:\

04/17/2022 07:34 PM <DIR> Autodesk

01/06/2021 05:05 PM <DIR> Drivers

02/16/2023 04:30 PM <DIR> Jsonnew

12/07/2019 02:44 PM <DIR> PerfLogs

02/16/2023 11:30 AM <DIR> Program Files

02/14/2023 03:54 PM <DIR> Program Files (x86)

07/15/2021 12:26 AM <DIR> SProV8i SS6

07/15/2021 12:59 AM <DIR> Staad.foundation 5.3

02/17/2023 10:27 AM <DIR> Tamil

04/25/2021 03:17 PM <DIR> Users

02/14/2023 04:12 PM <DIR> Windows

04/18/2021 08:31 PM <DIR> Windows10Upgrade

08/26/2022 06:58 PM <DIR> xampp

0 File(s) 0 bytes

13 Dir(s) 383,494,348,800 bytes free

C:\>E:

E:\nodejs>type .env

The system cannot find the file specified.

E:\nodejs>type port.env

The system cannot find the file specified.

E:\nodejs>port.env

'port.env' is not recognized as an internal or external command,

operable program or batch file.

E:\nodejs>node axios1.js

E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825

AxiosError.call(axiosError, error.message, code, config, request, response);

^

AxiosError: connect ECONNREFUSED ::1:8080

at AxiosError.from (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825:14)

at RedirectableRequest.handleRequestError (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:2965:25)

at RedirectableRequest.emit (node:events:513:28)

at eventHandlers.<computed> (E:\nodejs\node\_modules\follow-redirects\index.js:14:24)

at ClientRequest.emit (node:events:513:28)

at Socket.socketErrorListener (node:\_http\_client:502:9)

at Socket.emit (node:events:513:28)

at emitErrorNT (node:internal/streams/destroy:151:8)

at emitErrorCloseNT (node:internal/streams/destroy:116:3)

at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) {

port: 8080,

address: '::1',

syscall: 'connect',

code: 'ECONNREFUSED',

errno: -4078,

config: {

transitional: {

silentJSONParsing: true,

forcedJSONParsing: true,

clarifyTimeoutError: false

},

adapter: [ 'xhr', 'http' ],

transformRequest: [ [Function: transformRequest] ],

transformResponse: [ [Function: transformResponse] ],

timeout: 0,

xsrfCookieName: 'XSRF-TOKEN',

xsrfHeaderName: 'X-XSRF-TOKEN',

maxContentLength: -1,

maxBodyLength: -1,

env: {

FormData: [Function: FormData] {

LINE\_BREAK: '\r\n',

DEFAULT\_CONTENT\_TYPE: 'application/octet-stream'

},

Blob: [class Blob]

},

validateStatus: [Function: validateStatus],

headers: AxiosHeaders {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

method: 'post',

url: 'http://localhost:8080/',

data: undefined

},

request: <ref \*2> Writable {

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: false,

decodeStrings: true,

defaultEncoding: 'utf8',

length: 0,

writing: false,

corked: 0,

sync: true,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: null,

writelen: 0,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 0,

constructed: true,

prefinished: false,

errorEmitted: false,

emitClose: true,

autoDestroy: true,

errored: null,

closed: false,

closeEmitted: false,

[Symbol(kOnFinished)]: []

},

\_events: [Object: null prototype] {

response: [Function: handleResponse],

error: [Function: handleRequestError],

socket: [Function: handleRequestSocket]

},

\_eventsCount: 3,

\_maxListeners: undefined,

\_options: {

maxRedirects: 21,

maxBodyLength: Infinity,

protocol: 'http:',

path: '/',

method: 'POST',

headers: [Object: null prototype] {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

agents: { http: undefined, https: undefined },

auth: undefined,

beforeRedirect: [Function: dispatchBeforeRedirect],

beforeRedirects: { proxy: [Function: beforeRedirect] },

hostname: 'localhost',

port: '8080',

agent: undefined,

nativeProtocols: {

'http:': {

\_connectionListener: [Function: connectionListener],

METHODS: [

'ACL', 'BIND', 'CHECKOUT',

'CONNECT', 'COPY', 'DELETE',

'GET', 'HEAD', 'LINK',

'LOCK', 'M-SEARCH', 'MERGE',

'MKACTIVITY', 'MKCALENDAR', 'MKCOL',

'MOVE', 'NOTIFY', 'OPTIONS',

'PATCH', 'POST', 'PROPFIND',

'PROPPATCH', 'PURGE', 'PUT',

'REBIND', 'REPORT', 'SEARCH',

'SOURCE', 'SUBSCRIBE', 'TRACE',

'UNBIND', 'UNLINK', 'UNLOCK',

'UNSUBSCRIBE'

],

STATUS\_CODES: {

'100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'103': 'Early Hints',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'208': 'Already Reported',

'226': 'IM Used',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Found',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'308': 'Permanent Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Timeout',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Payload Too Large',

'414': 'URI Too Long',

'415': 'Unsupported Media Type',

'416': 'Range Not Satisfiable',

'417': 'Expectation Failed',

'418': "I'm a Teapot",

'421': 'Misdirected Request',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Too Early',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'451': 'Unavailable For Legal Reasons',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Timeout',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'508': 'Loop Detected',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required'

},

Agent: [Function: Agent] { defaultMaxSockets: Infinity },

ClientRequest: [Function: ClientRequest],

IncomingMessage: [Function: IncomingMessage],

OutgoingMessage: [Function: OutgoingMessage],

Server: [Function: Server],

ServerResponse: [Function: ServerResponse],

createServer: [Function: createServer],

validateHeaderName: [Function: \_\_node\_internal\_],

validateHeaderValue: [Function: \_\_node\_internal\_],

get: [Function: get],

request: [Function: request],

setMaxIdleHTTPParsers: [Function: setMaxIdleHTTPParsers],

maxHeaderSize: [Getter],

globalAgent: [Getter/Setter]

},

'https:': {

Agent: [Function: Agent],

globalAgent: Agent {

\_events: [Object: null prototype],

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 443,

protocol: 'https:',

options: [Object: null prototype],

requests: [Object: null prototype] {},

sockets: [Object: null prototype] {},

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 0,

maxCachedSessions: 100,

\_sessionCache: [Object],

[Symbol(kCapture)]: false

},

Server: [Function: Server],

createServer: [Function: createServer],

get: [Function: get],

request: [Function: request]

}

},

pathname: '/'

},

\_ended: true,

\_ending: true,

\_redirectCount: 0,

\_redirects: [],

\_requestBodyLength: 0,

\_requestBodyBuffers: [],

\_onNativeResponse: [Function (anonymous)],

\_currentRequest: <ref \*1> ClientRequest {

\_events: [Object: null prototype] {

response: [Function: bound onceWrapper] {

listener: [Function (anonymous)]

},

abort: [Function (anonymous)],

aborted: [Function (anonymous)],

connect: [Function (anonymous)],

error: [Function (anonymous)],

socket: [Function (anonymous)],

timeout: [Function (anonymous)]

},

\_eventsCount: 7,

\_maxListeners: undefined,

outputData: [],

outputSize: 0,

writable: true,

destroyed: false,

\_last: true,

chunkedEncoding: false,

shouldKeepAlive: false,

maxRequestsOnConnectionReached: false,

\_defaultKeepAlive: true,

useChunkedEncodingByDefault: true,

sendDate: false,

\_removedConnection: false,

\_removedContLen: false,

\_removedTE: false,

strictContentLength: false,

\_contentLength: 0,

\_hasBody: true,

\_trailer: '',

finished: true,

\_headerSent: true,

\_closed: false,

socket: Socket {

connecting: false,

\_hadError: true,

\_parent: null,

\_host: 'localhost',

\_closeAfterHandlingError: false,

\_readableState: ReadableState {

objectMode: false,

highWaterMark: 16384,

buffer: BufferList { head: null, tail: null, length: 0 },

length: 0,

pipes: [],

flowing: true,

ended: false,

endEmitted: false,

reading: true,

constructed: true,

sync: false,

needReadable: true,

emittedReadable: false,

readableListening: false,

resumeScheduled: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

destroyed: true,

errored: Error: connect ECONNREFUSED ::1:8080

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 8080

},

closed: true,

closeEmitted: true,

defaultEncoding: 'utf8',

awaitDrainWriters: null,

multiAwaitDrain: false,

readingMore: false,

dataEmitted: false,

decoder: null,

encoding: null,

[Symbol(kPaused)]: false

},

\_events: [Object: null prototype] {

end: [Function: onReadableStreamEnd],

connect: [ [Function], [Function], [Function] ],

free: [Function: onFree],

close: [

[Function: onClose],

[Function: socketCloseListener],

[Function]

],

timeout: [Function: onTimeout],

agentRemove: [Function: onRemove],

error: [Function: socketErrorListener],

drain: [Function: ondrain]

},

\_eventsCount: 8,

\_maxListeners: undefined,

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: true,

decodeStrings: false,

defaultEncoding: 'utf8',

length: 242,

writing: true,

corked: 0,

sync: false,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: [Function: bound onFinish],

writelen: 242,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 1,

constructed: true,

prefinished: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

errored: Error: connect ECONNREFUSED ::1:8080

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 8080

},

closed: true,

closeEmitted: true,

[Symbol(kOnFinished)]: []

},

allowHalfOpen: false,

\_sockname: null,

\_pendingData: 'POST / HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:8080\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_pendingEncoding: 'latin1',

server: null,

\_server: null,

parser: null,

\_httpMessage: [Circular \*1],

[Symbol(async\_id\_symbol)]: 4,

[Symbol(kHandle)]: null,

[Symbol(lastWriteQueueSize)]: 0,

[Symbol(timeout)]: null,

[Symbol(kBuffer)]: null,

[Symbol(kBufferCb)]: null,

[Symbol(kBufferGen)]: null,

[Symbol(kCapture)]: false,

[Symbol(kSetNoDelay)]: true,

[Symbol(kSetKeepAlive)]: true,

[Symbol(kSetKeepAliveInitialDelay)]: 60,

[Symbol(kBytesRead)]: 0,

[Symbol(kBytesWritten)]: 0

},

\_header: 'POST / HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:8080\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_keepAliveTimeout: 0,

\_onPendingData: [Function: nop],

agent: Agent {

\_events: [Object: null prototype] {

free: [Function (anonymous)],

newListener: [Function: maybeEnableKeylog]

},

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 80,

protocol: 'http:',

options: [Object: null prototype] { noDelay: true, path: null },

requests: [Object: null prototype] {},

sockets: [Object: null prototype] { 'localhost:8080:': [ [Socket] ] },

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 1,

[Symbol(kCapture)]: false

},

socketPath: undefined,

method: 'POST',

maxHeaderSize: undefined,

insecureHTTPParser: undefined,

joinDuplicateHeaders: undefined,

path: '/',

\_ended: false,

res: null,

aborted: false,

timeoutCb: null,

upgradeOrConnect: false,

parser: null,

maxHeadersCount: null,

reusedSocket: false,

host: 'localhost',

protocol: 'http:',

\_redirectable: [Circular \*2],

[Symbol(kCapture)]: false,

[Symbol(kBytesWritten)]: 0,

[Symbol(kEndCalled)]: true,

[Symbol(kNeedDrain)]: false,

[Symbol(corked)]: 0,

[Symbol(kOutHeaders)]: [Object: null prototype] {

accept: [ 'Accept', 'application/json, text/plain, \*/\*' ],

'content-type': [ 'Content-Type', 'application/x-www-form-urlencoded' ],

'user-agent': [ 'User-Agent', 'axios/1.3.3' ],

'accept-encoding': [ 'Accept-Encoding', 'gzip, compress, deflate, br' ],

host: [ 'Host', 'localhost:8080' ]

},

[Symbol(errored)]: null,

[Symbol(kUniqueHeaders)]: null

},

\_currentUrl: 'http://localhost:8080/',

[Symbol(kCapture)]: false

},

cause: Error: connect ECONNREFUSED ::1:8080

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 8080

}

}

Node.js v18.14.0

E:\nodejs>

C:\Users\admin\vue-project\1\2\node\_modules\shell-quote\test

C:\WINDOWS\system32>cd C:\Tamil\mongosh-1.7.1-win32-x64\mongosh-1.7.1-win32-x64\bin

C:\Tamil\mongosh-1.7.1-win32-x64\mongosh-1.7.1-win32-x64\bin>mongosh.exe

Current Mongosh Log ID: 63edcb5b099ccd46772d9988

Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.7.1

MongoNetworkError: connect ECONNREFUSED 127.0.0.1:27017

C:\Tamil\mongosh-1.7.1-win32-x64\mongosh-1.7.1-win32-x64\bin>mongosh "mongodb+srv://cluster0.lbwtqdt.mongodb.net/myFirstDatabase" --apiVersion 1 --username tamilinfo87

Enter password: \*\*\*\*\*\*\*\*\*\*\*

Current Mongosh Log ID: 63edcbb61d85257bccc88a21

Connecting to: mongodb+srv://<credentials>@cluster0.lbwtqdt.mongodb.net/myFirstDatabase?appName=mongosh+1.7.1

Using MongoDB: 5.0.14 (API Version 1)

Using Mongosh: 1.7.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> show collections

MyCollection

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> show databases

myFirstDatabase 8.00 KiB

admin 336.00 KiB

local 9.15 GiB

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.createCollection("posts")

{ ok: 1 }

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> show collections

MyCollection

posts

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.posts.insertOne({"title": "Post 1"})

{

acknowledged: true,

insertedId: ObjectId("63edcc441f99465709d0d379")

}

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.posts.insertOne({

... title: "Post Title 1",

... body: "Body of post.",

... category: "News",

... likes: 1,

... tags: ["news", "events"],

... date: Date()

... })

{

acknowledged: true,

insertedId: ObjectId("63edcc5a1f99465709d0d37a")

}

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.posts.insertMany([

... {

... title: "Post Title 2",

... body: "Body of post.",

... category: "Event",

... likes: 2,

... tags: ["news", "events"],

... date: Date()

... },

... {

... title: "Post Title 3",

... body: "Body of post.",

... category: "Technology",

... likes: 3,

... tags: ["news", "events"],

... date: Date()

... },

... {

... title: "Post Title 4",

... body: "Body of post.",

... category: "Event",

... likes: 4,

... tags: ["news", "events"],

... date: Date()

... }

... ])

{

acknowledged: true,

insertedIds: {

'0': ObjectId("63edcc721f99465709d0d37b"),

'1': ObjectId("63edcc721f99465709d0d37c"),

'2': ObjectId("63edcc721f99465709d0d37d")

}

}

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.posts.find()

[

{ \_id: ObjectId("63edcc441f99465709d0d379"), title: 'Post 1' },

{

\_id: ObjectId("63edcc5a1f99465709d0d37a"),

title: 'Post Title 1',

body: 'Body of post.',

category: 'News',

likes: 1,

tags: [ 'news', 'events' ],

date: 'Thu Feb 16 2023 11:55:30 GMT+0530 (India Standard Time)'

},

{

\_id: ObjectId("63edcc721f99465709d0d37b"),

title: 'Post Title 2',

body: 'Body of post.',

category: 'Event',

likes: 2,

tags: [ 'news', 'events' ],

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

},

{

\_id: ObjectId("63edcc721f99465709d0d37c"),

title: 'Post Title 3',

body: 'Body of post.',

category: 'Technology',

likes: 3,

tags: [ 'news', 'events' ],

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

},

{

\_id: ObjectId("63edcc721f99465709d0d37d"),

title: 'Post Title 4',

body: 'Body of post.',

category: 'Event',

likes: 4,

tags: [ 'news', 'events' ],

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

}

]

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.posts.find({}, {title: 1, date: 1})

[

{ \_id: ObjectId("63edcc441f99465709d0d379"), title: 'Post 1' },

{

\_id: ObjectId("63edcc5a1f99465709d0d37a"),

title: 'Post Title 1',

date: 'Thu Feb 16 2023 11:55:30 GMT+0530 (India Standard Time)'

},

{

\_id: ObjectId("63edcc721f99465709d0d37b"),

title: 'Post Title 2',

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

},

{

\_id: ObjectId("63edcc721f99465709d0d37c"),

title: 'Post Title 3',

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

},

{

\_id: ObjectId("63edcc721f99465709d0d37d"),

title: 'Post Title 4',

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

}

]

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.posts.find({}, {\_id: 0, title: 1, date: 1})

[

{ title: 'Post 1' },

{

title: 'Post Title 1',

date: 'Thu Feb 16 2023 11:55:30 GMT+0530 (India Standard Time)'

},

{

title: 'Post Title 2',

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

},

{

title: 'Post Title 3',

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

},

{

title: 'Post Title 4',

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

}

]

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.posts.find({}, {category: 0})

[

{ \_id: ObjectId("63edcc441f99465709d0d379"), title: 'Post 1' },

{

\_id: ObjectId("63edcc5a1f99465709d0d37a"),

title: 'Post Title 1',

body: 'Body of post.',

likes: 1,

tags: [ 'news', 'events' ],

date: 'Thu Feb 16 2023 11:55:30 GMT+0530 (India Standard Time)'

},

{

\_id: ObjectId("63edcc721f99465709d0d37b"),

title: 'Post Title 2',

body: 'Body of post.',

likes: 2,

tags: [ 'news', 'events' ],

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

},

{

\_id: ObjectId("63edcc721f99465709d0d37c"),

title: 'Post Title 3',

body: 'Body of post.',

likes: 3,

tags: [ 'news', 'events' ],

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

},

{

\_id: ObjectId("63edcc721f99465709d0d37d"),

title: 'Post Title 4',

body: 'Body of post.',

likes: 4,

tags: [ 'news', 'events' ],

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

}

]

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.posts.find( { title: "Post Title 1" } )

[

{

\_id: ObjectId("63edcc5a1f99465709d0d37a"),

title: 'Post Title 1',

body: 'Body of post.',

category: 'News',

likes: 1,

tags: [ 'news', 'events' ],

date: 'Thu Feb 16 2023 11:55:30 GMT+0530 (India Standard Time)'

}

]

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.posts.find({}, {\_id: 0, title: 1, date: 1,likes:1})

[

{ title: 'Post 1' },

{nsertedId: null,

title: 'Post Title 1',

likes: 12,t: 1,

date: 'Thu Feb 16 2023 11:55:30 GMT+0530 (India Standard Time)'

},

{

title: 'Post Title 2',

likes: 2,

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

},

{

title: 'Post Title 3',

likes: 3,

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

},

{

title: 'Post Title 4',

likes: 4,

date: 'Thu Feb 16 2023 11:55:54 GMT+0530 (India Standard Time)'

}

]

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.posts.deleteOne({ title: "Post Title 1" })

{ acknowledged: true, deletedCount: 1 }

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> $eq: Values are equal

Uncaught:

SyntaxError: Missing semicolon. (1:11)

> 1 | $eq: Values are equal

| ^

2 |

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.posts.aggregate([

... {

... $match: { likes: { $gt: 1 } }

... },

... {

... $group: { \_id: "$category", totalLikes: { $sum: "$likes" } }

... }

... ])

[

{ \_id: 'Event', totalLikes: 6 },

{ \_id: 'Technology', totalLikes: 3 }

]

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> db.posts.aggregate([

... // Stage 1: Only find documents that have more than 1 like

... {

... $match: { likes: { $gt: 1 } }

... },

... // Stage 2: Group documents by category and sum each categories likes

... {

... $group: { \_id: "$category", totalLikes: { $sum: "$likes" } }

... }

... ])

[

{ \_id: 'Technology', totalLikes: 3 },

{ \_id: 'Event', totalLikes: 6 }

]

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> show collections

MyCollection

posts

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase> show databases

myFirstDatabase 80.00 KiB

admin 336.00 KiB

local 9.63 GiB

Atlas atlas-chdvyr-shard-0 [primary] myFirstDatabase>

Microsoft Windows [Version 10.0.19044.2364]

(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>e:

E:\>cd nodejs

E:\nodejs>node axios5.js

E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825

AxiosError.call(axiosError, error.message, code, config, request, response);

^

AxiosError: connect ECONNREFUSED ::1:5000

at AxiosError.from (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825:14)

at RedirectableRequest.handleRequestError (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:2965:25)

at RedirectableRequest.emit (node:events:513:28)

at eventHandlers.<computed> (E:\nodejs\node\_modules\follow-redirects\index.js:14:24)

at ClientRequest.emit (node:events:513:28)

at Socket.socketErrorListener (node:\_http\_client:502:9)

at Socket.emit (node:events:513:28)

at emitErrorNT (node:internal/streams/destroy:151:8)

at emitErrorCloseNT (node:internal/streams/destroy:116:3)

at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) {

port: 5000,

address: '::1',

syscall: 'connect',

code: 'ECONNREFUSED',

errno: -4078,

config: {

transitional: {

silentJSONParsing: true,

forcedJSONParsing: true,

clarifyTimeoutError: false

},

adapter: [ 'xhr', 'http' ],

transformRequest: [ [Function: transformRequest] ],

transformResponse: [ [Function: transformResponse] ],

timeout: 0,

xsrfCookieName: 'XSRF-TOKEN',

xsrfHeaderName: 'X-XSRF-TOKEN',

maxContentLength: -1,

maxBodyLength: -1,

env: {

FormData: [Function: FormData] {

LINE\_BREAK: '\r\n',

DEFAULT\_CONTENT\_TYPE: 'application/octet-stream'

},

Blob: [class Blob]

},

validateStatus: [Function: validateStatus],

headers: AxiosHeaders {

Accept: 'application/json, text/plain, \*/\*',

Authorization: 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

method: 'get',

url: 'http://localhost:5000/user/validateToken',

data: undefined

},

request: <ref \*2> Writable {

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: false,

decodeStrings: true,

defaultEncoding: 'utf8',

length: 0,

writing: false,

corked: 0,

sync: true,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: null,

writelen: 0,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 0,

constructed: true,

prefinished: false,

errorEmitted: false,

emitClose: true,

autoDestroy: true,

errored: null,

closed: false,

closeEmitted: false,

[Symbol(kOnFinished)]: []

},

\_events: [Object: null prototype] {

response: [Function: handleResponse],

error: [Function: handleRequestError],

socket: [Function: handleRequestSocket]

},

\_eventsCount: 3,

\_maxListeners: undefined,

\_options: {

maxRedirects: 21,

maxBodyLength: Infinity,

protocol: 'http:',

path: '/user/validateToken',

method: 'GET',

headers: [Object: null prototype] {

Accept: 'application/json, text/plain, \*/\*',

Authorization: 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

agents: { http: undefined, https: undefined },

auth: undefined,

beforeRedirect: [Function: dispatchBeforeRedirect],

beforeRedirects: { proxy: [Function: beforeRedirect] },

hostname: 'localhost',

port: '5000',

agent: undefined,

nativeProtocols: {

'http:': {

\_connectionListener: [Function: connectionListener],

METHODS: [

'ACL', 'BIND', 'CHECKOUT',

'CONNECT', 'COPY', 'DELETE',

'GET', 'HEAD', 'LINK',

'LOCK', 'M-SEARCH', 'MERGE',

'MKACTIVITY', 'MKCALENDAR', 'MKCOL',

'MOVE', 'NOTIFY', 'OPTIONS',

'PATCH', 'POST', 'PROPFIND',

'PROPPATCH', 'PURGE', 'PUT',

'REBIND', 'REPORT', 'SEARCH',

'SOURCE', 'SUBSCRIBE', 'TRACE',

'UNBIND', 'UNLINK', 'UNLOCK',

'UNSUBSCRIBE'

],

STATUS\_CODES: {

'100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'103': 'Early Hints',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'208': 'Already Reported',

'226': 'IM Used',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Found',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'308': 'Permanent Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Timeout',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Payload Too Large',

'414': 'URI Too Long',

'415': 'Unsupported Media Type',

'416': 'Range Not Satisfiable',

'417': 'Expectation Failed',

'418': "I'm a Teapot",

'421': 'Misdirected Request',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Too Early',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'451': 'Unavailable For Legal Reasons',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Timeout',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'508': 'Loop Detected',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required'

},

Agent: [Function: Agent] { defaultMaxSockets: Infinity },

ClientRequest: [Function: ClientRequest],

IncomingMessage: [Function: IncomingMessage],

OutgoingMessage: [Function: OutgoingMessage],

Server: [Function: Server],

ServerResponse: [Function: ServerResponse],

createServer: [Function: createServer],

validateHeaderName: [Function: \_\_node\_internal\_],

validateHeaderValue: [Function: \_\_node\_internal\_],

get: [Function: get],

request: [Function: request],

setMaxIdleHTTPParsers: [Function: setMaxIdleHTTPParsers],

maxHeaderSize: [Getter],

globalAgent: [Getter/Setter]

},

'https:': {

Agent: [Function: Agent],

globalAgent: Agent {

\_events: [Object: null prototype],

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 443,

protocol: 'https:',

options: [Object: null prototype],

requests: [Object: null prototype] {},

sockets: [Object: null prototype] {},

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 0,

maxCachedSessions: 100,

\_sessionCache: [Object],

[Symbol(kCapture)]: false

},

Server: [Function: Server],

createServer: [Function: createServer],

get: [Function: get],

request: [Function: request]

}

},

pathname: '/user/validateToken'

},

\_ended: true,

\_ending: true,

\_redirectCount: 0,

\_redirects: [],

\_requestBodyLength: 0,

\_requestBodyBuffers: [],

\_onNativeResponse: [Function (anonymous)],

\_currentRequest: <ref \*1> ClientRequest {

\_events: [Object: null prototype] {

response: [Function: bound onceWrapper] {

listener: [Function (anonymous)]

},

abort: [Function (anonymous)],

aborted: [Function (anonymous)],

connect: [Function (anonymous)],

error: [Function (anonymous)],

socket: [Function (anonymous)],

timeout: [Function (anonymous)]

},

\_eventsCount: 7,

\_maxListeners: undefined,

outputData: [],

outputSize: 0,

writable: true,

destroyed: false,

\_last: true,

chunkedEncoding: false,

shouldKeepAlive: false,

maxRequestsOnConnectionReached: false,

\_defaultKeepAlive: true,

useChunkedEncodingByDefault: false,

sendDate: false,

\_removedConnection: false,

\_removedContLen: false,

\_removedTE: false,

strictContentLength: false,

\_contentLength: 0,

\_hasBody: true,

\_trailer: '',

finished: true,

\_headerSent: true,

\_closed: false,

socket: Socket {

connecting: false,

\_hadError: true,

\_parent: null,

\_host: 'localhost',

\_closeAfterHandlingError: false,

\_readableState: ReadableState {

objectMode: false,

highWaterMark: 16384,

buffer: BufferList { head: null, tail: null, length: 0 },

length: 0,

pipes: [],

flowing: true,

ended: false,

endEmitted: false,

reading: true,

constructed: true,

sync: false,

needReadable: true,

emittedReadable: false,

readableListening: false,

resumeScheduled: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

destroyed: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

defaultEncoding: 'utf8',

awaitDrainWriters: null,

multiAwaitDrain: false,

readingMore: false,

dataEmitted: false,

decoder: null,

encoding: null,

[Symbol(kPaused)]: false

},

\_events: [Object: null prototype] {

end: [Function: onReadableStreamEnd],

connect: [ [Function], [Function], [Function] ],

free: [Function: onFree],

close: [

[Function: onClose],

[Function: socketCloseListener],

[Function]

],

timeout: [Function: onTimeout],

agentRemove: [Function: onRemove],

error: [Function: socketErrorListener],

drain: [Function: ondrain]

},

\_eventsCount: 8,

\_maxListeners: undefined,

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: true,

decodeStrings: false,

defaultEncoding: 'utf8',

length: 423,

writing: true,

corked: 0,

sync: false,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: [Function: bound onFinish],

writelen: 423,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 1,

constructed: true,

prefinished: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

[Symbol(kOnFinished)]: []

},

allowHalfOpen: false,

\_sockname: null,

\_pendingData: 'GET /user/validateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'\r\n',

\_pendingEncoding: 'latin1',

server: null,

\_server: null,

parser: null,

\_httpMessage: [Circular \*1],

[Symbol(async\_id\_symbol)]: 4,

[Symbol(kHandle)]: null,

[Symbol(lastWriteQueueSize)]: 0,

[Symbol(timeout)]: null,

[Symbol(kBuffer)]: null,

[Symbol(kBufferCb)]: null,

[Symbol(kBufferGen)]: null,

[Symbol(kCapture)]: false,

[Symbol(kSetNoDelay)]: true,

[Symbol(kSetKeepAlive)]: true,

[Symbol(kSetKeepAliveInitialDelay)]: 60,

[Symbol(kBytesRead)]: 0,

[Symbol(kBytesWritten)]: 0

},

\_header: 'GET /user/validateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'\r\n',

\_keepAliveTimeout: 0,

\_onPendingData: [Function: nop],

agent: Agent {

\_events: [Object: null prototype] {

free: [Function (anonymous)],

newListener: [Function: maybeEnableKeylog]

},

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 80,

protocol: 'http:',

options: [Object: null prototype] { noDelay: true, path: null },

requests: [Object: null prototype] {},

sockets: [Object: null prototype] { 'localhost:5000:': [ [Socket] ] },

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 1,

[Symbol(kCapture)]: false

},

socketPath: undefined,

method: 'GET',

maxHeaderSize: undefined,

insecureHTTPParser: undefined,

joinDuplicateHeaders: undefined,

path: '/user/validateToken',

\_ended: false,

res: null,

aborted: false,

timeoutCb: null,

upgradeOrConnect: false,

parser: null,

maxHeadersCount: null,

reusedSocket: false,

host: 'localhost',

protocol: 'http:',

\_redirectable: [Circular \*2],

[Symbol(kCapture)]: false,

[Symbol(kBytesWritten)]: 0,

[Symbol(kEndCalled)]: true,

[Symbol(kNeedDrain)]: false,

[Symbol(corked)]: 0,

[Symbol(kOutHeaders)]: [Object: null prototype] {

accept: [ 'Accept', 'application/json, text/plain, \*/\*' ],

authorization: [

'Authorization',

'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA'

],

'user-agent': [ 'User-Agent', 'axios/1.3.3' ],

'accept-encoding': [ 'Accept-Encoding', 'gzip, compress, deflate, br' ],

host: [ 'Host', 'localhost:5000' ]

},

[Symbol(errored)]: null,

[Symbol(kUniqueHeaders)]: null

},

\_currentUrl: 'http://localhost:5000/user/validateToken',

[Symbol(kCapture)]: false

},

cause: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

}

}

Node.js v18.14.0

E:\nodejs>node axios5.js

E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825

AxiosError.call(axiosError, error.message, code, config, request, response);

^

AxiosError: connect ECONNREFUSED ::1:5001

at AxiosError.from (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825:14)

at RedirectableRequest.handleRequestError (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:2965:25)

at RedirectableRequest.emit (node:events:513:28)

at eventHandlers.<computed> (E:\nodejs\node\_modules\follow-redirects\index.js:14:24)

at ClientRequest.emit (node:events:513:28)

at Socket.socketErrorListener (node:\_http\_client:502:9)

at Socket.emit (node:events:513:28)

at emitErrorNT (node:internal/streams/destroy:151:8)

at emitErrorCloseNT (node:internal/streams/destroy:116:3)

at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) {

port: 5001,

address: '::1',

syscall: 'connect',

code: 'ECONNREFUSED',

errno: -4078,

config: {

transitional: {

silentJSONParsing: true,

forcedJSONParsing: true,

clarifyTimeoutError: false

},

adapter: [ 'xhr', 'http' ],

transformRequest: [ [Function: transformRequest] ],

transformResponse: [ [Function: transformResponse] ],

timeout: 0,

xsrfCookieName: 'XSRF-TOKEN',

xsrfHeaderName: 'X-XSRF-TOKEN',

maxContentLength: -1,

maxBodyLength: -1,

env: {

FormData: [Function: FormData] {

LINE\_BREAK: '\r\n',

DEFAULT\_CONTENT\_TYPE: 'application/octet-stream'

},

Blob: [class Blob]

},

validateStatus: [Function: validateStatus],

headers: AxiosHeaders {

Accept: 'application/json, text/plain, \*/\*',

Authorization: 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

method: 'get',

url: 'http://localhost:5001/user/validateToken',

data: undefined

},

request: <ref \*2> Writable {

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: false,

decodeStrings: true,

defaultEncoding: 'utf8',

length: 0,

writing: false,

corked: 0,

sync: true,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: null,

writelen: 0,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 0,

constructed: true,

prefinished: false,

errorEmitted: false,

emitClose: true,

autoDestroy: true,

errored: null,

closed: false,

closeEmitted: false,

[Symbol(kOnFinished)]: []

},

\_events: [Object: null prototype] {

response: [Function: handleResponse],

error: [Function: handleRequestError],

socket: [Function: handleRequestSocket]

},

\_eventsCount: 3,

\_maxListeners: undefined,

\_options: {

maxRedirects: 21,

maxBodyLength: Infinity,

protocol: 'http:',

path: '/user/validateToken',

method: 'GET',

headers: [Object: null prototype] {

Accept: 'application/json, text/plain, \*/\*',

Authorization: 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

agents: { http: undefined, https: undefined },

auth: undefined,

beforeRedirect: [Function: dispatchBeforeRedirect],

beforeRedirects: { proxy: [Function: beforeRedirect] },

hostname: 'localhost',

port: '5001',

agent: undefined,

nativeProtocols: {

'http:': {

\_connectionListener: [Function: connectionListener],

METHODS: [

'ACL', 'BIND', 'CHECKOUT',

'CONNECT', 'COPY', 'DELETE',

'GET', 'HEAD', 'LINK',

'LOCK', 'M-SEARCH', 'MERGE',

'MKACTIVITY', 'MKCALENDAR', 'MKCOL',

'MOVE', 'NOTIFY', 'OPTIONS',

'PATCH', 'POST', 'PROPFIND',

'PROPPATCH', 'PURGE', 'PUT',

'REBIND', 'REPORT', 'SEARCH',

'SOURCE', 'SUBSCRIBE', 'TRACE',

'UNBIND', 'UNLINK', 'UNLOCK',

'UNSUBSCRIBE'

],

STATUS\_CODES: {

'100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'103': 'Early Hints',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'208': 'Already Reported',

'226': 'IM Used',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Found',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'308': 'Permanent Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Timeout',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Payload Too Large',

'414': 'URI Too Long',

'415': 'Unsupported Media Type',

'416': 'Range Not Satisfiable',

'417': 'Expectation Failed',

'418': "I'm a Teapot",

'421': 'Misdirected Request',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Too Early',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'451': 'Unavailable For Legal Reasons',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Timeout',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'508': 'Loop Detected',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required'

},

Agent: [Function: Agent] { defaultMaxSockets: Infinity },

ClientRequest: [Function: ClientRequest],

IncomingMessage: [Function: IncomingMessage],

OutgoingMessage: [Function: OutgoingMessage],

Server: [Function: Server],

ServerResponse: [Function: ServerResponse],

createServer: [Function: createServer],

validateHeaderName: [Function: \_\_node\_internal\_],

validateHeaderValue: [Function: \_\_node\_internal\_],

get: [Function: get],

request: [Function: request],

setMaxIdleHTTPParsers: [Function: setMaxIdleHTTPParsers],

maxHeaderSize: [Getter],

globalAgent: [Getter/Setter]

},

'https:': {

Agent: [Function: Agent],

globalAgent: Agent {

\_events: [Object: null prototype],

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 443,

protocol: 'https:',

options: [Object: null prototype],

requests: [Object: null prototype] {},

sockets: [Object: null prototype] {},

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 0,

maxCachedSessions: 100,

\_sessionCache: [Object],

[Symbol(kCapture)]: false

},

Server: [Function: Server],

createServer: [Function: createServer],

get: [Function: get],

request: [Function: request]

}

},

pathname: '/user/validateToken'

},

\_ended: true,

\_ending: true,

\_redirectCount: 0,

\_redirects: [],

\_requestBodyLength: 0,

\_requestBodyBuffers: [],

\_onNativeResponse: [Function (anonymous)],

\_currentRequest: <ref \*1> ClientRequest {

\_events: [Object: null prototype] {

response: [Function: bound onceWrapper] {

listener: [Function (anonymous)]

},

abort: [Function (anonymous)],

aborted: [Function (anonymous)],

connect: [Function (anonymous)],

error: [Function (anonymous)],

socket: [Function (anonymous)],

timeout: [Function (anonymous)]

},

\_eventsCount: 7,

\_maxListeners: undefined,

outputData: [],

outputSize: 0,

writable: true,

destroyed: false,

\_last: true,

chunkedEncoding: false,

shouldKeepAlive: false,

maxRequestsOnConnectionReached: false,

\_defaultKeepAlive: true,

useChunkedEncodingByDefault: false,

sendDate: false,

\_removedConnection: false,

\_removedContLen: false,

\_removedTE: false,

strictContentLength: false,

\_contentLength: 0,

\_hasBody: true,

\_trailer: '',

finished: true,

\_headerSent: true,

\_closed: false,

socket: Socket {

connecting: false,

\_hadError: true,

\_parent: null,

\_host: 'localhost',

\_closeAfterHandlingError: false,

\_readableState: ReadableState {

objectMode: false,

highWaterMark: 16384,

buffer: BufferList { head: null, tail: null, length: 0 },

length: 0,

pipes: [],

flowing: true,

ended: false,

endEmitted: false,

reading: true,

constructed: true,

sync: false,

needReadable: true,

emittedReadable: false,

readableListening: false,

resumeScheduled: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

destroyed: true,

errored: Error: connect ECONNREFUSED ::1:5001

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5001

},

closed: true,

closeEmitted: true,

defaultEncoding: 'utf8',

awaitDrainWriters: null,

multiAwaitDrain: false,

readingMore: false,

dataEmitted: false,

decoder: null,

encoding: null,

[Symbol(kPaused)]: false

},

\_events: [Object: null prototype] {

end: [Function: onReadableStreamEnd],

connect: [ [Function], [Function], [Function] ],

free: [Function: onFree],

close: [

[Function: onClose],

[Function: socketCloseListener],

[Function]

],

timeout: [Function: onTimeout],

agentRemove: [Function: onRemove],

error: [Function: socketErrorListener],

drain: [Function: ondrain]

},

\_eventsCount: 8,

\_maxListeners: undefined,

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: true,

decodeStrings: false,

defaultEncoding: 'utf8',

length: 423,

writing: true,

corked: 0,

sync: false,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: [Function: bound onFinish],

writelen: 423,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 1,

constructed: true,

prefinished: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

errored: Error: connect ECONNREFUSED ::1:5001

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5001

},

closed: true,

closeEmitted: true,

[Symbol(kOnFinished)]: []

},

allowHalfOpen: false,

\_sockname: null,

\_pendingData: 'GET /user/validateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5001\r\n' +

'Connection: close\r\n' +

'\r\n',

\_pendingEncoding: 'latin1',

server: null,

\_server: null,

parser: null,

\_httpMessage: [Circular \*1],

[Symbol(async\_id\_symbol)]: 4,

[Symbol(kHandle)]: null,

[Symbol(lastWriteQueueSize)]: 0,

[Symbol(timeout)]: null,

[Symbol(kBuffer)]: null,

[Symbol(kBufferCb)]: null,

[Symbol(kBufferGen)]: null,

[Symbol(kCapture)]: false,

[Symbol(kSetNoDelay)]: true,

[Symbol(kSetKeepAlive)]: true,

[Symbol(kSetKeepAliveInitialDelay)]: 60,

[Symbol(kBytesRead)]: 0,

[Symbol(kBytesWritten)]: 0

},

\_header: 'GET /user/validateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5001\r\n' +

'Connection: close\r\n' +

'\r\n',

\_keepAliveTimeout: 0,

\_onPendingData: [Function: nop],

agent: Agent {

\_events: [Object: null prototype] {

free: [Function (anonymous)],

newListener: [Function: maybeEnableKeylog]

},

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 80,

protocol: 'http:',

options: [Object: null prototype] { noDelay: true, path: null },

requests: [Object: null prototype] {},

sockets: [Object: null prototype] { 'localhost:5001:': [ [Socket] ] },

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 1,

[Symbol(kCapture)]: false

},

socketPath: undefined,

method: 'GET',

maxHeaderSize: undefined,

insecureHTTPParser: undefined,

joinDuplicateHeaders: undefined,

path: '/user/validateToken',

\_ended: false,

res: null,

aborted: false,

timeoutCb: null,

upgradeOrConnect: false,

parser: null,

maxHeadersCount: null,

reusedSocket: false,

host: 'localhost',

protocol: 'http:',

\_redirectable: [Circular \*2],

[Symbol(kCapture)]: false,

[Symbol(kBytesWritten)]: 0,

[Symbol(kEndCalled)]: true,

[Symbol(kNeedDrain)]: false,

[Symbol(corked)]: 0,

[Symbol(kOutHeaders)]: [Object: null prototype] {

accept: [ 'Accept', 'application/json, text/plain, \*/\*' ],

authorization: [

'Authorization',

'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA'

],

'user-agent': [ 'User-Agent', 'axios/1.3.3' ],

'accept-encoding': [ 'Accept-Encoding', 'gzip, compress, deflate, br' ],

host: [ 'Host', 'localhost:5001' ]

},

[Symbol(errored)]: null,

[Symbol(kUniqueHeaders)]: null

},

\_currentUrl: 'http://localhost:5001/user/validateToken',

[Symbol(kCapture)]: false

},

cause: Error: connect ECONNREFUSED ::1:5001

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5001

}

}

Node.js v18.14.0

E:\nodejs>node axios1.js

E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825

AxiosError.call(axiosError, error.message, code, config, request, response);

^

AxiosError: connect ECONNREFUSED ::1:8080

at AxiosError.from (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825:14)

at RedirectableRequest.handleRequestError (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:2965:25)

at RedirectableRequest.emit (node:events:513:28)

at eventHandlers.<computed> (E:\nodejs\node\_modules\follow-redirects\index.js:14:24)

at ClientRequest.emit (node:events:513:28)

at Socket.socketErrorListener (node:\_http\_client:502:9)

at Socket.emit (node:events:513:28)

at emitErrorNT (node:internal/streams/destroy:151:8)

at emitErrorCloseNT (node:internal/streams/destroy:116:3)

at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) {

port: 8080,

address: '::1',

syscall: 'connect',

code: 'ECONNREFUSED',

errno: -4078,

config: {

transitional: {

silentJSONParsing: true,

forcedJSONParsing: true,

clarifyTimeoutError: false

},

adapter: [ 'xhr', 'http' ],

transformRequest: [ [Function: transformRequest] ],

transformResponse: [ [Function: transformResponse] ],

timeout: 0,

xsrfCookieName: 'XSRF-TOKEN',

xsrfHeaderName: 'X-XSRF-TOKEN',

maxContentLength: -1,

maxBodyLength: -1,

env: {

FormData: [Function: FormData] {

LINE\_BREAK: '\r\n',

DEFAULT\_CONTENT\_TYPE: 'application/octet-stream'

},

Blob: [class Blob]

},

validateStatus: [Function: validateStatus],

headers: AxiosHeaders {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

method: 'post',

url: 'http://localhost:8080/',

data: undefined

},

request: <ref \*2> Writable {

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: false,

decodeStrings: true,

defaultEncoding: 'utf8',

length: 0,

writing: false,

corked: 0,

sync: true,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: null,

writelen: 0,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 0,

constructed: true,

prefinished: false,

errorEmitted: false,

emitClose: true,

autoDestroy: true,

errored: null,

closed: false,

closeEmitted: false,

[Symbol(kOnFinished)]: []

},

\_events: [Object: null prototype] {

response: [Function: handleResponse],

error: [Function: handleRequestError],

socket: [Function: handleRequestSocket]

},

\_eventsCount: 3,

\_maxListeners: undefined,

\_options: {

maxRedirects: 21,

maxBodyLength: Infinity,

protocol: 'http:',

path: '/',

method: 'POST',

headers: [Object: null prototype] {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

agents: { http: undefined, https: undefined },

auth: undefined,

beforeRedirect: [Function: dispatchBeforeRedirect],

beforeRedirects: { proxy: [Function: beforeRedirect] },

hostname: 'localhost',

port: '8080',

agent: undefined,

nativeProtocols: {

'http:': {

\_connectionListener: [Function: connectionListener],

METHODS: [

'ACL', 'BIND', 'CHECKOUT',

'CONNECT', 'COPY', 'DELETE',

'GET', 'HEAD', 'LINK',

'LOCK', 'M-SEARCH', 'MERGE',

'MKACTIVITY', 'MKCALENDAR', 'MKCOL',

'MOVE', 'NOTIFY', 'OPTIONS',

'PATCH', 'POST', 'PROPFIND',

'PROPPATCH', 'PURGE', 'PUT',

'REBIND', 'REPORT', 'SEARCH',

'SOURCE', 'SUBSCRIBE', 'TRACE',

'UNBIND', 'UNLINK', 'UNLOCK',

'UNSUBSCRIBE'

],

STATUS\_CODES: {

'100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'103': 'Early Hints',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'208': 'Already Reported',

'226': 'IM Used',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Found',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'308': 'Permanent Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Timeout',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Payload Too Large',

'414': 'URI Too Long',

'415': 'Unsupported Media Type',

'416': 'Range Not Satisfiable',

'417': 'Expectation Failed',

'418': "I'm a Teapot",

'421': 'Misdirected Request',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Too Early',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'451': 'Unavailable For Legal Reasons',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Timeout',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'508': 'Loop Detected',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required'

},

Agent: [Function: Agent] { defaultMaxSockets: Infinity },

ClientRequest: [Function: ClientRequest],

IncomingMessage: [Function: IncomingMessage],

OutgoingMessage: [Function: OutgoingMessage],

Server: [Function: Server],

ServerResponse: [Function: ServerResponse],

createServer: [Function: createServer],

validateHeaderName: [Function: \_\_node\_internal\_],

validateHeaderValue: [Function: \_\_node\_internal\_],

get: [Function: get],

request: [Function: request],

setMaxIdleHTTPParsers: [Function: setMaxIdleHTTPParsers],

maxHeaderSize: [Getter],

globalAgent: [Getter/Setter]

},

'https:': {

Agent: [Function: Agent],

globalAgent: Agent {

\_events: [Object: null prototype],

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 443,

protocol: 'https:',

options: [Object: null prototype],

requests: [Object: null prototype] {},

sockets: [Object: null prototype] {},

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 0,

maxCachedSessions: 100,

\_sessionCache: [Object],

[Symbol(kCapture)]: false

},

Server: [Function: Server],

createServer: [Function: createServer],

get: [Function: get],

request: [Function: request]

}

},

pathname: '/'

},

\_ended: true,

\_ending: true,

\_redirectCount: 0,

\_redirects: [],

\_requestBodyLength: 0,

\_requestBodyBuffers: [],

\_onNativeResponse: [Function (anonymous)],

\_currentRequest: <ref \*1> ClientRequest {

\_events: [Object: null prototype] {

response: [Function: bound onceWrapper] {

listener: [Function (anonymous)]

},

abort: [Function (anonymous)],

aborted: [Function (anonymous)],

connect: [Function (anonymous)],

error: [Function (anonymous)],

socket: [Function (anonymous)],

timeout: [Function (anonymous)]

},

\_eventsCount: 7,

\_maxListeners: undefined,

outputData: [],

outputSize: 0,

writable: true,

destroyed: false,

\_last: true,

chunkedEncoding: false,

shouldKeepAlive: false,

maxRequestsOnConnectionReached: false,

\_defaultKeepAlive: true,

useChunkedEncodingByDefault: true,

sendDate: false,

\_removedConnection: false,

\_removedContLen: false,

\_removedTE: false,

strictContentLength: false,

\_contentLength: 0,

\_hasBody: true,

\_trailer: '',

finished: true,

\_headerSent: true,

\_closed: false,

socket: Socket {

connecting: false,

\_hadError: true,

\_parent: null,

\_host: 'localhost',

\_closeAfterHandlingError: false,

\_readableState: ReadableState {

objectMode: false,

highWaterMark: 16384,

buffer: BufferList { head: null, tail: null, length: 0 },

length: 0,

pipes: [],

flowing: true,

ended: false,

endEmitted: false,

reading: true,

constructed: true,

sync: false,

needReadable: true,

emittedReadable: false,

readableListening: false,

resumeScheduled: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

destroyed: true,

errored: Error: connect ECONNREFUSED ::1:8080

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 8080

},

closed: true,

closeEmitted: true,

defaultEncoding: 'utf8',

awaitDrainWriters: null,

multiAwaitDrain: false,

readingMore: false,

dataEmitted: false,

decoder: null,

encoding: null,

[Symbol(kPaused)]: false

},

\_events: [Object: null prototype] {

end: [Function: onReadableStreamEnd],

connect: [ [Function], [Function], [Function] ],

free: [Function: onFree],

close: [

[Function: onClose],

[Function: socketCloseListener],

[Function]

],

timeout: [Function: onTimeout],

agentRemove: [Function: onRemove],

error: [Function: socketErrorListener],

drain: [Function: ondrain]

},

\_eventsCount: 8,

\_maxListeners: undefined,

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: true,

decodeStrings: false,

defaultEncoding: 'utf8',

length: 242,

writing: true,

corked: 0,

sync: false,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: [Function: bound onFinish],

writelen: 242,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 1,

constructed: true,

prefinished: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

errored: Error: connect ECONNREFUSED ::1:8080

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 8080

},

closed: true,

closeEmitted: true,

[Symbol(kOnFinished)]: []

},

allowHalfOpen: false,

\_sockname: null,

\_pendingData: 'POST / HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:8080\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_pendingEncoding: 'latin1',

server: null,

\_server: null,

parser: null,

\_httpMessage: [Circular \*1],

[Symbol(async\_id\_symbol)]: 4,

[Symbol(kHandle)]: null,

[Symbol(lastWriteQueueSize)]: 0,

[Symbol(timeout)]: null,

[Symbol(kBuffer)]: null,

[Symbol(kBufferCb)]: null,

[Symbol(kBufferGen)]: null,

[Symbol(kCapture)]: false,

[Symbol(kSetNoDelay)]: true,

[Symbol(kSetKeepAlive)]: true,

[Symbol(kSetKeepAliveInitialDelay)]: 60,

[Symbol(kBytesRead)]: 0,

[Symbol(kBytesWritten)]: 0

},

\_header: 'POST / HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:8080\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_keepAliveTimeout: 0,

\_onPendingData: [Function: nop],

agent: Agent {

\_events: [Object: null prototype] {

free: [Function (anonymous)],

newListener: [Function: maybeEnableKeylog]

},

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 80,

protocol: 'http:',

options: [Object: null prototype] { noDelay: true, path: null },

requests: [Object: null prototype] {},

sockets: [Object: null prototype] { 'localhost:8080:': [ [Socket] ] },

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 1,

[Symbol(kCapture)]: false

},

socketPath: undefined,

method: 'POST',

maxHeaderSize: undefined,

insecureHTTPParser: undefined,

joinDuplicateHeaders: undefined,

path: '/',

\_ended: false,

res: null,

aborted: false,

timeoutCb: null,

upgradeOrConnect: false,

parser: null,

maxHeadersCount: null,

reusedSocket: false,

host: 'localhost',

protocol: 'http:',

\_redirectable: [Circular \*2],

[Symbol(kCapture)]: false,

[Symbol(kBytesWritten)]: 0,

[Symbol(kEndCalled)]: true,

[Symbol(kNeedDrain)]: false,

[Symbol(corked)]: 0,

[Symbol(kOutHeaders)]: [Object: null prototype] {

accept: [ 'Accept', 'application/json, text/plain, \*/\*' ],

'content-type': [ 'Content-Type', 'application/x-www-form-urlencoded' ],

'user-agent': [ 'User-Agent', 'axios/1.3.3' ],

'accept-encoding': [ 'Accept-Encoding', 'gzip, compress, deflate, br' ],

host: [ 'Host', 'localhost:8080' ]

},

[Symbol(errored)]: null,

[Symbol(kUniqueHeaders)]: null

},

\_currentUrl: 'http://localhost:8080/',

[Symbol(kCapture)]: false

},

cause: Error: connect ECONNREFUSED ::1:8080

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 8080

}

}

Node.js v18.14.0

E:\nodejs>node axios5.js

E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825

AxiosError.call(axiosError, error.message, code, config, request, response);

^

AxiosError: connect ECONNREFUSED ::1:5003

at AxiosError.from (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825:14)

at RedirectableRequest.handleRequestError (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:2965:25)

at RedirectableRequest.emit (node:events:513:28)

at eventHandlers.<computed> (E:\nodejs\node\_modules\follow-redirects\index.js:14:24)

at ClientRequest.emit (node:events:513:28)

at Socket.socketErrorListener (node:\_http\_client:502:9)

at Socket.emit (node:events:513:28)

at emitErrorNT (node:internal/streams/destroy:151:8)

at emitErrorCloseNT (node:internal/streams/destroy:116:3)

at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) {

port: 5003,

address: '::1',

syscall: 'connect',

code: 'ECONNREFUSED',

errno: -4078,

config: {

transitional: {

silentJSONParsing: true,

forcedJSONParsing: true,

clarifyTimeoutError: false

},

adapter: [ 'xhr', 'http' ],

transformRequest: [ [Function: transformRequest] ],

transformResponse: [ [Function: transformResponse] ],

timeout: 0,

xsrfCookieName: 'XSRF-TOKEN',

xsrfHeaderName: 'X-XSRF-TOKEN',

maxContentLength: -1,

maxBodyLength: -1,

env: {

FormData: [Function: FormData] {

LINE\_BREAK: '\r\n',

DEFAULT\_CONTENT\_TYPE: 'application/octet-stream'

},

Blob: [class Blob]

},

validateStatus: [Function: validateStatus],

headers: AxiosHeaders {

Accept: 'application/json, text/plain, \*/\*',

Authorization: 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

method: 'get',

url: 'http://localhost:5003/user/validateToken',

data: undefined

},

request: <ref \*2> Writable {

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: false,

decodeStrings: true,

defaultEncoding: 'utf8',

length: 0,

writing: false,

corked: 0,

sync: true,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: null,

writelen: 0,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 0,

constructed: true,

prefinished: false,

errorEmitted: false,

emitClose: true,

autoDestroy: true,

errored: null,

closed: false,

closeEmitted: false,

[Symbol(kOnFinished)]: []

},

\_events: [Object: null prototype] {

response: [Function: handleResponse],

error: [Function: handleRequestError],

socket: [Function: handleRequestSocket]

},

\_eventsCount: 3,

\_maxListeners: undefined,

\_options: {

maxRedirects: 21,

maxBodyLength: Infinity,

protocol: 'http:',

path: '/user/validateToken',

method: 'GET',

headers: [Object: null prototype] {

Accept: 'application/json, text/plain, \*/\*',

Authorization: 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

agents: { http: undefined, https: undefined },

auth: undefined,

beforeRedirect: [Function: dispatchBeforeRedirect],

beforeRedirects: { proxy: [Function: beforeRedirect] },

hostname: 'localhost',

port: '5003',

agent: undefined,

nativeProtocols: {

'http:': {

\_connectionListener: [Function: connectionListener],

METHODS: [

'ACL', 'BIND', 'CHECKOUT',

'CONNECT', 'COPY', 'DELETE',

'GET', 'HEAD', 'LINK',

'LOCK', 'M-SEARCH', 'MERGE',

'MKACTIVITY', 'MKCALENDAR', 'MKCOL',

'MOVE', 'NOTIFY', 'OPTIONS',

'PATCH', 'POST', 'PROPFIND',

'PROPPATCH', 'PURGE', 'PUT',

'REBIND', 'REPORT', 'SEARCH',

'SOURCE', 'SUBSCRIBE', 'TRACE',

'UNBIND', 'UNLINK', 'UNLOCK',

'UNSUBSCRIBE'

],

STATUS\_CODES: {

'100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'103': 'Early Hints',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'208': 'Already Reported',

'226': 'IM Used',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Found',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'308': 'Permanent Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Timeout',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Payload Too Large',

'414': 'URI Too Long',

'415': 'Unsupported Media Type',

'416': 'Range Not Satisfiable',

'417': 'Expectation Failed',

'418': "I'm a Teapot",

'421': 'Misdirected Request',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Too Early',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'451': 'Unavailable For Legal Reasons',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Timeout',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'508': 'Loop Detected',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required'

},

Agent: [Function: Agent] { defaultMaxSockets: Infinity },

ClientRequest: [Function: ClientRequest],

IncomingMessage: [Function: IncomingMessage],

OutgoingMessage: [Function: OutgoingMessage],

Server: [Function: Server],

ServerResponse: [Function: ServerResponse],

createServer: [Function: createServer],

validateHeaderName: [Function: \_\_node\_internal\_],

validateHeaderValue: [Function: \_\_node\_internal\_],

get: [Function: get],

request: [Function: request],

setMaxIdleHTTPParsers: [Function: setMaxIdleHTTPParsers],

maxHeaderSize: [Getter],

globalAgent: [Getter/Setter]

},

'https:': {

Agent: [Function: Agent],

globalAgent: Agent {

\_events: [Object: null prototype],

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 443,

protocol: 'https:',

options: [Object: null prototype],

requests: [Object: null prototype] {},

sockets: [Object: null prototype] {},

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 0,

maxCachedSessions: 100,

\_sessionCache: [Object],

[Symbol(kCapture)]: false

},

Server: [Function: Server],

createServer: [Function: createServer],

get: [Function: get],

request: [Function: request]

}

},

pathname: '/user/validateToken'

},

\_ended: true,

\_ending: true,

\_redirectCount: 0,

\_redirects: [],

\_requestBodyLength: 0,

\_requestBodyBuffers: [],

\_onNativeResponse: [Function (anonymous)],

\_currentRequest: <ref \*1> ClientRequest {

\_events: [Object: null prototype] {

response: [Function: bound onceWrapper] {

listener: [Function (anonymous)]

},

abort: [Function (anonymous)],

aborted: [Function (anonymous)],

connect: [Function (anonymous)],

error: [Function (anonymous)],

socket: [Function (anonymous)],

timeout: [Function (anonymous)]

},

\_eventsCount: 7,

\_maxListeners: undefined,

outputData: [],

outputSize: 0,

writable: true,

destroyed: false,

\_last: true,

chunkedEncoding: false,

shouldKeepAlive: false,

maxRequestsOnConnectionReached: false,

\_defaultKeepAlive: true,

useChunkedEncodingByDefault: false,

sendDate: false,

\_removedConnection: false,

\_removedContLen: false,

\_removedTE: false,

strictContentLength: false,

\_contentLength: 0,

\_hasBody: true,

\_trailer: '',

finished: true,

\_headerSent: true,

\_closed: false,

socket: Socket {

connecting: false,

\_hadError: true,

\_parent: null,

\_host: 'localhost',

\_closeAfterHandlingError: false,

\_readableState: ReadableState {

objectMode: false,

highWaterMark: 16384,

buffer: BufferList { head: null, tail: null, length: 0 },

length: 0,

pipes: [],

flowing: true,

ended: false,

endEmitted: false,

reading: true,

constructed: true,

sync: false,

needReadable: true,

emittedReadable: false,

readableListening: false,

resumeScheduled: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

destroyed: true,

errored: Error: connect ECONNREFUSED ::1:5003

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5003

},

closed: true,

closeEmitted: true,

defaultEncoding: 'utf8',

awaitDrainWriters: null,

multiAwaitDrain: false,

readingMore: false,

dataEmitted: false,

decoder: null,

encoding: null,

[Symbol(kPaused)]: false

},

\_events: [Object: null prototype] {

end: [Function: onReadableStreamEnd],

connect: [ [Function], [Function], [Function] ],

free: [Function: onFree],

close: [

[Function: onClose],

[Function: socketCloseListener],

[Function]

],

timeout: [Function: onTimeout],

agentRemove: [Function: onRemove],

error: [Function: socketErrorListener],

drain: [Function: ondrain]

},

\_eventsCount: 8,

\_maxListeners: undefined,

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: true,

decodeStrings: false,

defaultEncoding: 'utf8',

length: 423,

writing: true,

corked: 0,

sync: false,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: [Function: bound onFinish],

writelen: 423,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 1,

constructed: true,

prefinished: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

errored: Error: connect ECONNREFUSED ::1:5003

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5003

},

closed: true,

closeEmitted: true,

[Symbol(kOnFinished)]: []

},

allowHalfOpen: false,

\_sockname: null,

\_pendingData: 'GET /user/validateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5003\r\n' +

'Connection: close\r\n' +

'\r\n',

\_pendingEncoding: 'latin1',

server: null,

\_server: null,

parser: null,

\_httpMessage: [Circular \*1],

[Symbol(async\_id\_symbol)]: 4,

[Symbol(kHandle)]: null,

[Symbol(lastWriteQueueSize)]: 0,

[Symbol(timeout)]: null,

[Symbol(kBuffer)]: null,

[Symbol(kBufferCb)]: null,

[Symbol(kBufferGen)]: null,

[Symbol(kCapture)]: false,

[Symbol(kSetNoDelay)]: true,

[Symbol(kSetKeepAlive)]: true,

[Symbol(kSetKeepAliveInitialDelay)]: 60,

[Symbol(kBytesRead)]: 0,

[Symbol(kBytesWritten)]: 0

},

\_header: 'GET /user/validateToken HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5003\r\n' +

'Connection: close\r\n' +

'\r\n',

\_keepAliveTimeout: 0,

\_onPendingData: [Function: nop],

agent: Agent {

\_events: [Object: null prototype] {

free: [Function (anonymous)],

newListener: [Function: maybeEnableKeylog]

},

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 80,

protocol: 'http:',

options: [Object: null prototype] { noDelay: true, path: null },

requests: [Object: null prototype] {},

sockets: [Object: null prototype] { 'localhost:5003:': [ [Socket] ] },

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 1,

[Symbol(kCapture)]: false

},

socketPath: undefined,

method: 'GET',

maxHeaderSize: undefined,

insecureHTTPParser: undefined,

joinDuplicateHeaders: undefined,

path: '/user/validateToken',

\_ended: false,

res: null,

aborted: false,

timeoutCb: null,

upgradeOrConnect: false,

parser: null,

maxHeadersCount: null,

reusedSocket: false,

host: 'localhost',

protocol: 'http:',

\_redirectable: [Circular \*2],

[Symbol(kCapture)]: false,

[Symbol(kBytesWritten)]: 0,

[Symbol(kEndCalled)]: true,

[Symbol(kNeedDrain)]: false,

[Symbol(corked)]: 0,

[Symbol(kOutHeaders)]: [Object: null prototype] {

accept: [ 'Accept', 'application/json, text/plain, \*/\*' ],

authorization: [

'Authorization',

'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aW1lIjoiRnJpIEZlYiAxNyAyMDIzIDA3OjI4OjI0IEdNVCswNTMwIChJbmRpYSBTdGFuZGFyZCBUaW1lKSIsInVzZXJJZCI6MTIsImlhdCI6MTY3NjU5OTEwNH0.BZwZldFCyujM9uVuM4CO3BANDbj9KkvkKEAWA\_PXCMA'

],

'user-agent': [ 'User-Agent', 'axios/1.3.3' ],

'accept-encoding': [ 'Accept-Encoding', 'gzip, compress, deflate, br' ],

host: [ 'Host', 'localhost:5003' ]

},

[Symbol(errored)]: null,

[Symbol(kUniqueHeaders)]: null

},

\_currentUrl: 'http://localhost:5003/user/validateToken',

[Symbol(kCapture)]: false

},

cause: Error: connect ECONNREFUSED ::1:5003

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5003

}

}

Node.js v18.14.0

E:\nodejs>

Microsoft Windows [Version 10.0.19044.2364]

(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>e:

E:\>cd nodejs

E:\nodejs>curl google.com

<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">

<TITLE>301 Moved</TITLE></HEAD><BODY>

<H1>301 Moved</H1>

The document has moved

<A HREF="http://www.google.com/">here</A>.

</BODY></HTML>

E:\nodejs>node axios1.js

E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825

AxiosError.call(axiosError, error.message, code, config, request, response);

^

AxiosError: connect ECONNREFUSED ::1:8080

at AxiosError.from (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825:14)

at RedirectableRequest.handleRequestError (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:2965:25)

at RedirectableRequest.emit (node:events:513:28)

at eventHandlers.<computed> (E:\nodejs\node\_modules\follow-redirects\index.js:14:24)

at ClientRequest.emit (node:events:513:28)

at Socket.socketErrorListener (node:\_http\_client:502:9)

at Socket.emit (node:events:513:28)

at emitErrorNT (node:internal/streams/destroy:151:8)

at emitErrorCloseNT (node:internal/streams/destroy:116:3)

at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) {

port: 8080,

address: '::1',

syscall: 'connect',

code: 'ECONNREFUSED',

errno: -4078,

config: {

transitional: {

silentJSONParsing: true,

forcedJSONParsing: true,

clarifyTimeoutError: false

},

adapter: [ 'xhr', 'http' ],

transformRequest: [ [Function: transformRequest] ],

transformResponse: [ [Function: transformResponse] ],

timeout: 0,

xsrfCookieName: 'XSRF-TOKEN',

xsrfHeaderName: 'X-XSRF-TOKEN',

maxContentLength: -1,

maxBodyLength: -1,

env: {

FormData: [Function: FormData] {

LINE\_BREAK: '\r\n',

DEFAULT\_CONTENT\_TYPE: 'application/octet-stream'

},

Blob: [class Blob]

},

validateStatus: [Function: validateStatus],

headers: AxiosHeaders {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

method: 'post',

url: 'http://localhost:8080/',

data: undefined

},

request: <ref \*2> Writable {

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: false,

decodeStrings: true,

defaultEncoding: 'utf8',

length: 0,

writing: false,

corked: 0,

sync: true,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: null,

writelen: 0,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 0,

constructed: true,

prefinished: false,

errorEmitted: false,

emitClose: true,

autoDestroy: true,

errored: null,

closed: false,

closeEmitted: false,

[Symbol(kOnFinished)]: []

},

\_events: [Object: null prototype] {

response: [Function: handleResponse],

error: [Function: handleRequestError],

socket: [Function: handleRequestSocket]

},

\_eventsCount: 3,

\_maxListeners: undefined,

\_options: {

maxRedirects: 21,

maxBodyLength: Infinity,

protocol: 'http:',

path: '/',

method: 'POST',

headers: [Object: null prototype] {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

agents: { http: undefined, https: undefined },

auth: undefined,

beforeRedirect: [Function: dispatchBeforeRedirect],

beforeRedirects: { proxy: [Function: beforeRedirect] },

hostname: 'localhost',

port: '8080',

agent: undefined,

nativeProtocols: {

'http:': {

\_connectionListener: [Function: connectionListener],

METHODS: [

'ACL', 'BIND', 'CHECKOUT',

'CONNECT', 'COPY', 'DELETE',

'GET', 'HEAD', 'LINK',

'LOCK', 'M-SEARCH', 'MERGE',

'MKACTIVITY', 'MKCALENDAR', 'MKCOL',

'MOVE', 'NOTIFY', 'OPTIONS',

'PATCH', 'POST', 'PROPFIND',

'PROPPATCH', 'PURGE', 'PUT',

'REBIND', 'REPORT', 'SEARCH',

'SOURCE', 'SUBSCRIBE', 'TRACE',

'UNBIND', 'UNLINK', 'UNLOCK',

'UNSUBSCRIBE'

],

STATUS\_CODES: {

'100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'103': 'Early Hints',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'208': 'Already Reported',

'226': 'IM Used',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Found',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'308': 'Permanent Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Timeout',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Payload Too Large',

'414': 'URI Too Long',

'415': 'Unsupported Media Type',

'416': 'Range Not Satisfiable',

'417': 'Expectation Failed',

'418': "I'm a Teapot",

'421': 'Misdirected Request',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Too Early',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'451': 'Unavailable For Legal Reasons',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Timeout',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'508': 'Loop Detected',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required'

},

Agent: [Function: Agent] { defaultMaxSockets: Infinity },

ClientRequest: [Function: ClientRequest],

IncomingMessage: [Function: IncomingMessage],

OutgoingMessage: [Function: OutgoingMessage],

Server: [Function: Server],

ServerResponse: [Function: ServerResponse],

createServer: [Function: createServer],

validateHeaderName: [Function: \_\_node\_internal\_],

validateHeaderValue: [Function: \_\_node\_internal\_],

get: [Function: get],

request: [Function: request],

setMaxIdleHTTPParsers: [Function: setMaxIdleHTTPParsers],

maxHeaderSize: [Getter],

globalAgent: [Getter/Setter]

},

'https:': {

Agent: [Function: Agent],

globalAgent: Agent {

\_events: [Object: null prototype],

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 443,

protocol: 'https:',

options: [Object: null prototype],

requests: [Object: null prototype] {},

sockets: [Object: null prototype] {},

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 0,

maxCachedSessions: 100,

\_sessionCache: [Object],

[Symbol(kCapture)]: false

},

Server: [Function: Server],

createServer: [Function: createServer],

get: [Function: get],

request: [Function: request]

}

},

pathname: '/'

},

\_ended: true,

\_ending: true,

\_redirectCount: 0,

\_redirects: [],

\_requestBodyLength: 0,

\_requestBodyBuffers: [],

\_onNativeResponse: [Function (anonymous)],

\_currentRequest: <ref \*1> ClientRequest {

\_events: [Object: null prototype] {

response: [Function: bound onceWrapper] {

listener: [Function (anonymous)]

},

abort: [Function (anonymous)],

aborted: [Function (anonymous)],

connect: [Function (anonymous)],

error: [Function (anonymous)],

socket: [Function (anonymous)],

timeout: [Function (anonymous)]

},

\_eventsCount: 7,

\_maxListeners: undefined,

outputData: [],

outputSize: 0,

writable: true,

destroyed: false,

\_last: true,

chunkedEncoding: false,

shouldKeepAlive: false,

maxRequestsOnConnectionReached: false,

\_defaultKeepAlive: true,

useChunkedEncodingByDefault: true,

sendDate: false,

\_removedConnection: false,

\_removedContLen: false,

\_removedTE: false,

strictContentLength: false,

\_contentLength: 0,

\_hasBody: true,

\_trailer: '',

finished: true,

\_headerSent: true,

\_closed: false,

socket: Socket {

connecting: false,

\_hadError: true,

\_parent: null,

\_host: 'localhost',

\_closeAfterHandlingError: false,

\_readableState: ReadableState {

objectMode: false,

highWaterMark: 16384,

buffer: BufferList { head: null, tail: null, length: 0 },

length: 0,

pipes: [],

flowing: true,

ended: false,

endEmitted: false,

reading: true,

constructed: true,

sync: false,

needReadable: true,

emittedReadable: false,

readableListening: false,

resumeScheduled: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

destroyed: true,

errored: Error: connect ECONNREFUSED ::1:8080

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 8080

},

closed: true,

closeEmitted: true,

defaultEncoding: 'utf8',

awaitDrainWriters: null,

multiAwaitDrain: false,

readingMore: false,

dataEmitted: false,

decoder: null,

encoding: null,

[Symbol(kPaused)]: false

},

\_events: [Object: null prototype] {

end: [Function: onReadableStreamEnd],

connect: [ [Function], [Function], [Function] ],

free: [Function: onFree],

close: [

[Function: onClose],

[Function: socketCloseListener],

[Function]

],

timeout: [Function: onTimeout],

agentRemove: [Function: onRemove],

error: [Function: socketErrorListener],

drain: [Function: ondrain]

},

\_eventsCount: 8,

\_maxListeners: undefined,

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: true,

decodeStrings: false,

defaultEncoding: 'utf8',

length: 242,

writing: true,

corked: 0,

sync: false,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: [Function: bound onFinish],

writelen: 242,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 1,

constructed: true,

prefinished: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

errored: Error: connect ECONNREFUSED ::1:8080

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 8080

},

closed: true,

closeEmitted: true,

[Symbol(kOnFinished)]: []

},

allowHalfOpen: false,

\_sockname: null,

\_pendingData: 'POST / HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:8080\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_pendingEncoding: 'latin1',

server: null,

\_server: null,

parser: null,

\_httpMessage: [Circular \*1],

[Symbol(async\_id\_symbol)]: 4,

[Symbol(kHandle)]: null,

[Symbol(lastWriteQueueSize)]: 0,

[Symbol(timeout)]: null,

[Symbol(kBuffer)]: null,

[Symbol(kBufferCb)]: null,

[Symbol(kBufferGen)]: null,

[Symbol(kCapture)]: false,

[Symbol(kSetNoDelay)]: true,

[Symbol(kSetKeepAlive)]: true,

[Symbol(kSetKeepAliveInitialDelay)]: 60,

[Symbol(kBytesRead)]: 0,

[Symbol(kBytesWritten)]: 0

},

\_header: 'POST / HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:8080\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_keepAliveTimeout: 0,

\_onPendingData: [Function: nop],

agent: Agent {

\_events: [Object: null prototype] {

free: [Function (anonymous)],

newListener: [Function: maybeEnableKeylog]

},

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 80,

protocol: 'http:',

options: [Object: null prototype] { noDelay: true, path: null },

requests: [Object: null prototype] {},

sockets: [Object: null prototype] { 'localhost:8080:': [ [Socket] ] },

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 1,

[Symbol(kCapture)]: false

},

socketPath: undefined,

method: 'POST',

maxHeaderSize: undefined,

insecureHTTPParser: undefined,

joinDuplicateHeaders: undefined,

path: '/',

\_ended: false,

res: null,

aborted: false,

timeoutCb: null,

upgradeOrConnect: false,

parser: null,

maxHeadersCount: null,

reusedSocket: false,

host: 'localhost',

protocol: 'http:',

\_redirectable: [Circular \*2],

[Symbol(kCapture)]: false,

[Symbol(kBytesWritten)]: 0,

[Symbol(kEndCalled)]: true,

[Symbol(kNeedDrain)]: false,

[Symbol(corked)]: 0,

[Symbol(kOutHeaders)]: [Object: null prototype] {

accept: [ 'Accept', 'application/json, text/plain, \*/\*' ],

'content-type': [ 'Content-Type', 'application/x-www-form-urlencoded' ],

'user-agent': [ 'User-Agent', 'axios/1.3.3' ],

'accept-encoding': [ 'Accept-Encoding', 'gzip, compress, deflate, br' ],

host: [ 'Host', 'localhost:8080' ]

},

[Symbol(errored)]: null,

[Symbol(kUniqueHeaders)]: null

},

\_currentUrl: 'http://localhost:8080/',

[Symbol(kCapture)]: false

},

cause: Error: connect ECONNREFUSED ::1:8080

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 8080

}

}

Node.js v18.14.0

E:\nodejs>node axios1.js

E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825

AxiosError.call(axiosError, error.message, code, config, request, response);

^

AxiosError: connect ECONNREFUSED ::1:5000

at AxiosError.from (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:825:14)

at RedirectableRequest.handleRequestError (E:\nodejs\node\_modules\axios\dist\node\axios.cjs:2965:25)

at RedirectableRequest.emit (node:events:513:28)

at eventHandlers.<computed> (E:\nodejs\node\_modules\follow-redirects\index.js:14:24)

at ClientRequest.emit (node:events:513:28)

at Socket.socketErrorListener (node:\_http\_client:502:9)

at Socket.emit (node:events:513:28)

at emitErrorNT (node:internal/streams/destroy:151:8)

at emitErrorCloseNT (node:internal/streams/destroy:116:3)

at process.processTicksAndRejections (node:internal/process/task\_queues:82:21) {

port: 5000,

address: '::1',

syscall: 'connect',

code: 'ECONNREFUSED',

errno: -4078,

config: {

transitional: {

silentJSONParsing: true,

forcedJSONParsing: true,

clarifyTimeoutError: false

},

adapter: [ 'xhr', 'http' ],

transformRequest: [ [Function: transformRequest] ],

transformResponse: [ [Function: transformResponse] ],

timeout: 0,

xsrfCookieName: 'XSRF-TOKEN',

xsrfHeaderName: 'X-XSRF-TOKEN',

maxContentLength: -1,

maxBodyLength: -1,

env: {

FormData: [Function: FormData] {

LINE\_BREAK: '\r\n',

DEFAULT\_CONTENT\_TYPE: 'application/octet-stream'

},

Blob: [class Blob]

},

validateStatus: [Function: validateStatus],

headers: AxiosHeaders {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

method: 'post',

url: 'http://localhost:5000/',

data: undefined

},

request: <ref \*2> Writable {

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: false,

decodeStrings: true,

defaultEncoding: 'utf8',

length: 0,

writing: false,

corked: 0,

sync: true,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: null,

writelen: 0,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 0,

constructed: true,

prefinished: false,

errorEmitted: false,

emitClose: true,

autoDestroy: true,

errored: null,

closed: false,

closeEmitted: false,

[Symbol(kOnFinished)]: []

},

\_events: [Object: null prototype] {

response: [Function: handleResponse],

error: [Function: handleRequestError],

socket: [Function: handleRequestSocket]

},

\_eventsCount: 3,

\_maxListeners: undefined,

\_options: {

maxRedirects: 21,

maxBodyLength: Infinity,

protocol: 'http:',

path: '/',

method: 'POST',

headers: [Object: null prototype] {

Accept: 'application/json, text/plain, \*/\*',

'Content-Type': 'application/x-www-form-urlencoded',

'User-Agent': 'axios/1.3.3',

'Accept-Encoding': 'gzip, compress, deflate, br'

},

agents: { http: undefined, https: undefined },

auth: undefined,

beforeRedirect: [Function: dispatchBeforeRedirect],

beforeRedirects: { proxy: [Function: beforeRedirect] },

hostname: 'localhost',

port: '5000',

agent: undefined,

nativeProtocols: {

'http:': {

\_connectionListener: [Function: connectionListener],

METHODS: [

'ACL', 'BIND', 'CHECKOUT',

'CONNECT', 'COPY', 'DELETE',

'GET', 'HEAD', 'LINK',

'LOCK', 'M-SEARCH', 'MERGE',

'MKACTIVITY', 'MKCALENDAR', 'MKCOL',

'MOVE', 'NOTIFY', 'OPTIONS',

'PATCH', 'POST', 'PROPFIND',

'PROPPATCH', 'PURGE', 'PUT',

'REBIND', 'REPORT', 'SEARCH',

'SOURCE', 'SUBSCRIBE', 'TRACE',

'UNBIND', 'UNLINK', 'UNLOCK',

'UNSUBSCRIBE'

],

STATUS\_CODES: {

'100': 'Continue',

'101': 'Switching Protocols',

'102': 'Processing',

'103': 'Early Hints',

'200': 'OK',

'201': 'Created',

'202': 'Accepted',

'203': 'Non-Authoritative Information',

'204': 'No Content',

'205': 'Reset Content',

'206': 'Partial Content',

'207': 'Multi-Status',

'208': 'Already Reported',

'226': 'IM Used',

'300': 'Multiple Choices',

'301': 'Moved Permanently',

'302': 'Found',

'303': 'See Other',

'304': 'Not Modified',

'305': 'Use Proxy',

'307': 'Temporary Redirect',

'308': 'Permanent Redirect',

'400': 'Bad Request',

'401': 'Unauthorized',

'402': 'Payment Required',

'403': 'Forbidden',

'404': 'Not Found',

'405': 'Method Not Allowed',

'406': 'Not Acceptable',

'407': 'Proxy Authentication Required',

'408': 'Request Timeout',

'409': 'Conflict',

'410': 'Gone',

'411': 'Length Required',

'412': 'Precondition Failed',

'413': 'Payload Too Large',

'414': 'URI Too Long',

'415': 'Unsupported Media Type',

'416': 'Range Not Satisfiable',

'417': 'Expectation Failed',

'418': "I'm a Teapot",

'421': 'Misdirected Request',

'422': 'Unprocessable Entity',

'423': 'Locked',

'424': 'Failed Dependency',

'425': 'Too Early',

'426': 'Upgrade Required',

'428': 'Precondition Required',

'429': 'Too Many Requests',

'431': 'Request Header Fields Too Large',

'451': 'Unavailable For Legal Reasons',

'500': 'Internal Server Error',

'501': 'Not Implemented',

'502': 'Bad Gateway',

'503': 'Service Unavailable',

'504': 'Gateway Timeout',

'505': 'HTTP Version Not Supported',

'506': 'Variant Also Negotiates',

'507': 'Insufficient Storage',

'508': 'Loop Detected',

'509': 'Bandwidth Limit Exceeded',

'510': 'Not Extended',

'511': 'Network Authentication Required'

},

Agent: [Function: Agent] { defaultMaxSockets: Infinity },

ClientRequest: [Function: ClientRequest],

IncomingMessage: [Function: IncomingMessage],

OutgoingMessage: [Function: OutgoingMessage],

Server: [Function: Server],

ServerResponse: [Function: ServerResponse],

createServer: [Function: createServer],

validateHeaderName: [Function: \_\_node\_internal\_],

validateHeaderValue: [Function: \_\_node\_internal\_],

get: [Function: get],

request: [Function: request],

setMaxIdleHTTPParsers: [Function: setMaxIdleHTTPParsers],

maxHeaderSize: [Getter],

globalAgent: [Getter/Setter]

},

'https:': {

Agent: [Function: Agent],

globalAgent: Agent {

\_events: [Object: null prototype],

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 443,

protocol: 'https:',

options: [Object: null prototype],

requests: [Object: null prototype] {},

sockets: [Object: null prototype] {},

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 0,

maxCachedSessions: 100,

\_sessionCache: [Object],

[Symbol(kCapture)]: false

},

Server: [Function: Server],

createServer: [Function: createServer],

get: [Function: get],

request: [Function: request]

}

},

pathname: '/'

},

\_ended: true,

\_ending: true,

\_redirectCount: 0,

\_redirects: [],

\_requestBodyLength: 0,

\_requestBodyBuffers: [],

\_onNativeResponse: [Function (anonymous)],

\_currentRequest: <ref \*1> ClientRequest {

\_events: [Object: null prototype] {

response: [Function: bound onceWrapper] {

listener: [Function (anonymous)]

},

abort: [Function (anonymous)],

aborted: [Function (anonymous)],

connect: [Function (anonymous)],

error: [Function (anonymous)],

socket: [Function (anonymous)],

timeout: [Function (anonymous)]

},

\_eventsCount: 7,

\_maxListeners: undefined,

outputData: [],

outputSize: 0,

writable: true,

destroyed: false,

\_last: true,

chunkedEncoding: false,

shouldKeepAlive: false,

maxRequestsOnConnectionReached: false,

\_defaultKeepAlive: true,

useChunkedEncodingByDefault: true,

sendDate: false,

\_removedConnection: false,

\_removedContLen: false,

\_removedTE: false,

strictContentLength: false,

\_contentLength: 0,

\_hasBody: true,

\_trailer: '',

finished: true,

\_headerSent: true,

\_closed: false,

socket: Socket {

connecting: false,

\_hadError: true,

\_parent: null,

\_host: 'localhost',

\_closeAfterHandlingError: false,

\_readableState: ReadableState {

objectMode: false,

highWaterMark: 16384,

buffer: BufferList { head: null, tail: null, length: 0 },

length: 0,

pipes: [],

flowing: true,

ended: false,

endEmitted: false,

reading: true,

constructed: true,

sync: false,

needReadable: true,

emittedReadable: false,

readableListening: false,

resumeScheduled: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

destroyed: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

defaultEncoding: 'utf8',

awaitDrainWriters: null,

multiAwaitDrain: false,

readingMore: false,

dataEmitted: false,

decoder: null,

encoding: null,

[Symbol(kPaused)]: false

},

\_events: [Object: null prototype] {

end: [Function: onReadableStreamEnd],

connect: [ [Function], [Function], [Function] ],

free: [Function: onFree],

close: [

[Function: onClose],

[Function: socketCloseListener],

[Function]

],

timeout: [Function: onTimeout],

agentRemove: [Function: onRemove],

error: [Function: socketErrorListener],

drain: [Function: ondrain]

},

\_eventsCount: 8,

\_maxListeners: undefined,

\_writableState: WritableState {

objectMode: false,

highWaterMark: 16384,

finalCalled: false,

needDrain: false,

ending: false,

ended: false,

finished: false,

destroyed: true,

decodeStrings: false,

defaultEncoding: 'utf8',

length: 242,

writing: true,

corked: 0,

sync: false,

bufferProcessing: false,

onwrite: [Function: bound onwrite],

writecb: [Function: bound onFinish],

writelen: 242,

afterWriteTickInfo: null,

buffered: [],

bufferedIndex: 0,

allBuffers: true,

allNoop: true,

pendingcb: 1,

constructed: true,

prefinished: false,

errorEmitted: true,

emitClose: false,

autoDestroy: true,

errored: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

},

closed: true,

closeEmitted: true,

[Symbol(kOnFinished)]: []

},

allowHalfOpen: false,

\_sockname: null,

\_pendingData: 'POST / HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_pendingEncoding: 'latin1',

server: null,

\_server: null,

parser: null,

\_httpMessage: [Circular \*1],

[Symbol(async\_id\_symbol)]: 4,

[Symbol(kHandle)]: null,

[Symbol(lastWriteQueueSize)]: 0,

[Symbol(timeout)]: null,

[Symbol(kBuffer)]: null,

[Symbol(kBufferCb)]: null,

[Symbol(kBufferGen)]: null,

[Symbol(kCapture)]: false,

[Symbol(kSetNoDelay)]: true,

[Symbol(kSetKeepAlive)]: true,

[Symbol(kSetKeepAliveInitialDelay)]: 60,

[Symbol(kBytesRead)]: 0,

[Symbol(kBytesWritten)]: 0

},

\_header: 'POST / HTTP/1.1\r\n' +

'Accept: application/json, text/plain, \*/\*\r\n' +

'Content-Type: application/x-www-form-urlencoded\r\n' +

'User-Agent: axios/1.3.3\r\n' +

'Accept-Encoding: gzip, compress, deflate, br\r\n' +

'Host: localhost:5000\r\n' +

'Connection: close\r\n' +

'Content-Length: 0\r\n' +

'\r\n',

\_keepAliveTimeout: 0,

\_onPendingData: [Function: nop],

agent: Agent {

\_events: [Object: null prototype] {

free: [Function (anonymous)],

newListener: [Function: maybeEnableKeylog]

},

\_eventsCount: 2,

\_maxListeners: undefined,

defaultPort: 80,

protocol: 'http:',

options: [Object: null prototype] { noDelay: true, path: null },

requests: [Object: null prototype] {},

sockets: [Object: null prototype] { 'localhost:5000:': [ [Socket] ] },

freeSockets: [Object: null prototype] {},

keepAliveMsecs: 1000,

keepAlive: false,

maxSockets: Infinity,

maxFreeSockets: 256,

scheduling: 'lifo',

maxTotalSockets: Infinity,

totalSocketCount: 1,

[Symbol(kCapture)]: false

},

socketPath: undefined,

method: 'POST',

maxHeaderSize: undefined,

insecureHTTPParser: undefined,

joinDuplicateHeaders: undefined,

path: '/',

\_ended: false,

res: null,

aborted: false,

timeoutCb: null,

upgradeOrConnect: false,

parser: null,

maxHeadersCount: null,

reusedSocket: false,

host: 'localhost',

protocol: 'http:',

\_redirectable: [Circular \*2],

[Symbol(kCapture)]: false,

[Symbol(kBytesWritten)]: 0,

[Symbol(kEndCalled)]: true,

[Symbol(kNeedDrain)]: false,

[Symbol(corked)]: 0,

[Symbol(kOutHeaders)]: [Object: null prototype] {

accept: [ 'Accept', 'application/json, text/plain, \*/\*' ],

'content-type': [ 'Content-Type', 'application/x-www-form-urlencoded' ],

'user-agent': [ 'User-Agent', 'axios/1.3.3' ],

'accept-encoding': [ 'Accept-Encoding', 'gzip, compress, deflate, br' ],

host: [ 'Host', 'localhost:5000' ]

},

[Symbol(errored)]: null,

[Symbol(kUniqueHeaders)]: null

},

\_currentUrl: 'http://localhost:5000/',

[Symbol(kCapture)]: false

},

cause: Error: connect ECONNREFUSED ::1:5000

at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1494:16) {

errno: -4078,

code: 'ECONNREFUSED',

syscall: 'connect',

address: '::1',

port: 5000

}

}

Node.js v18.14.0

E:\nodejs>