

8:03

642.4
LTE KB/s 5G 74%



Data Analytics & GenAI...



Book now

Sponsored · codingninjas.c...

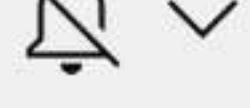
Introduction to Express Js | Sigma Web Development Course - Tutorial #88

160K views 1y ago Sigma Web Development Course - We... [more](#)

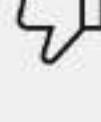


CodeWithHarry 8.29M

Join



Like 5.2K



Share

Remix

Download

Comments 725



I have recently learned HTML and CSS through Apna college paid course. I must say that the concepts they a...

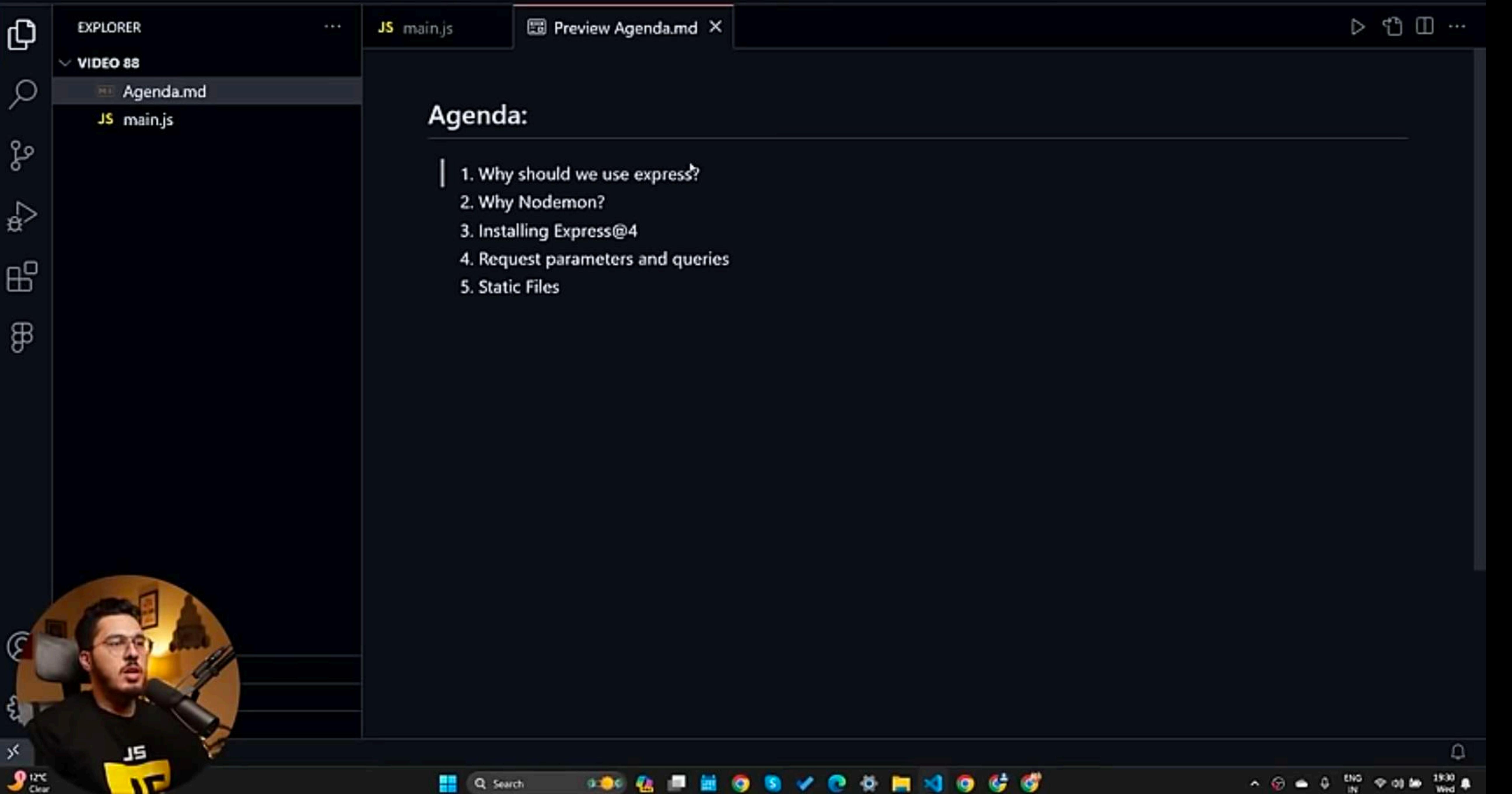


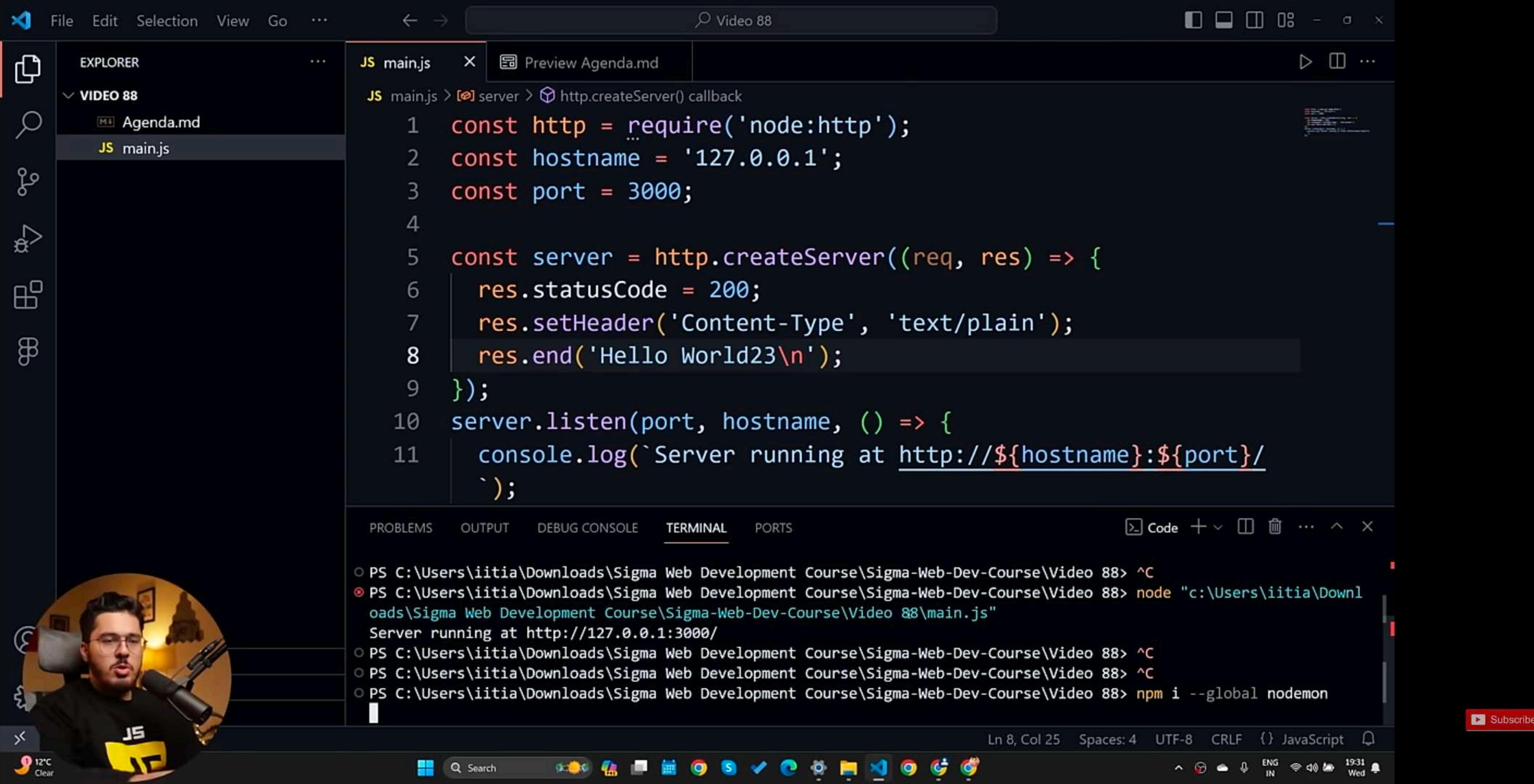
End of playlist

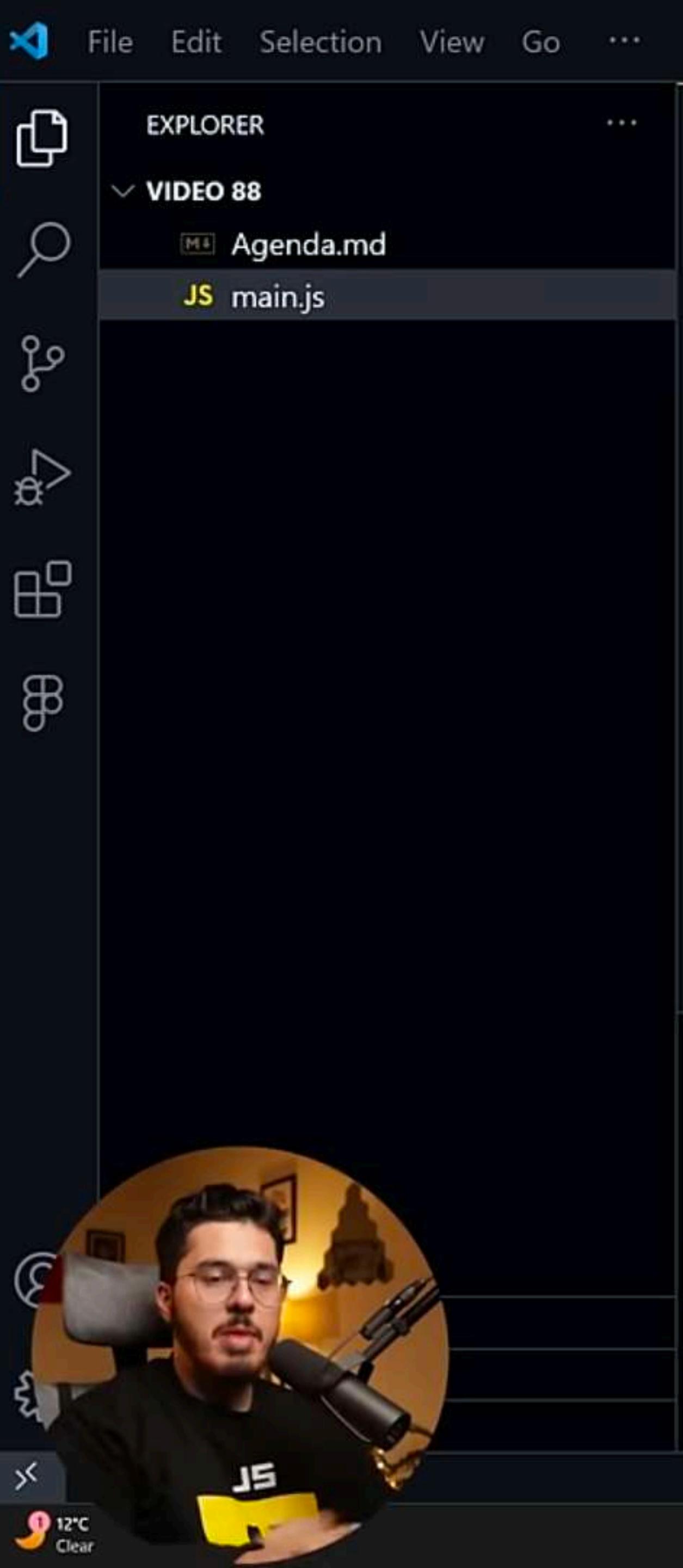


Today's Special • 4/4









File Edit Selection View Go ... ← → Video 88 □ □ □ 00 - □ ×

EXPLORER ... JS main.js X Preview Agenda.md

VIDEO 88

Agenda.md

JS main.js

```
JS main.js > [x] server > http.createServer() callback
1 const http = require('node:http');
2 const hostname = '127.0.0.1';
3 const port = 3000;
4
5 const server = http.createServer((req, res) => {
6   res.statusCode = 200;
7   res.setHeader('Content-Type', 'text/plain');
8   res.end('Hello World23\n');
9 });
10 server.listen(port, hostname, () => {
11   console.log(`Server running at http://${hostname}:${port}/
`);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Code + ×

```
changed 33 packages, and audited 34 packages in 2s
3 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 88> npm init -y
```

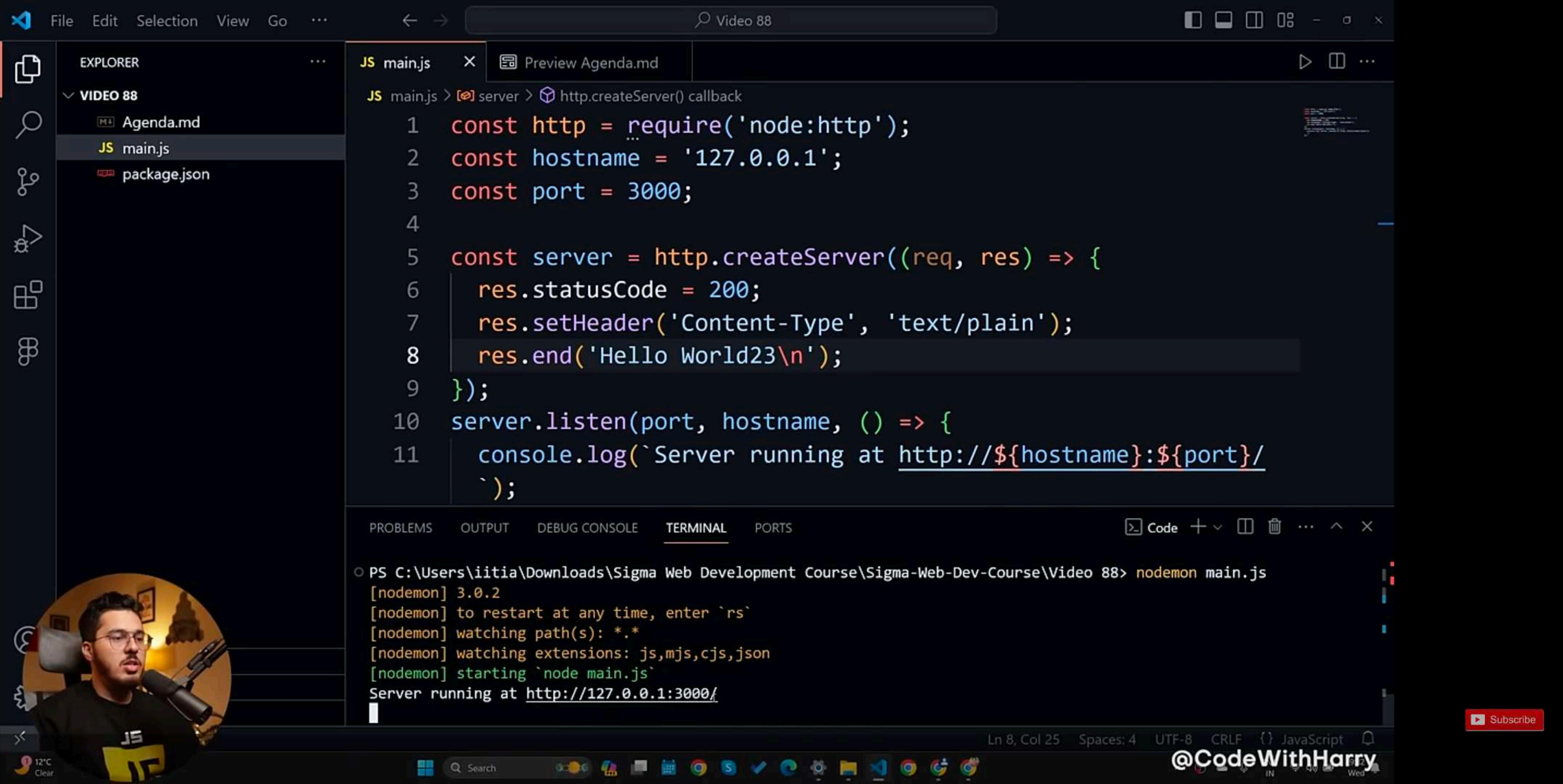
Ln 8, Col 25 Spaces: 4 UTF-8 CRLF {} JavaScript

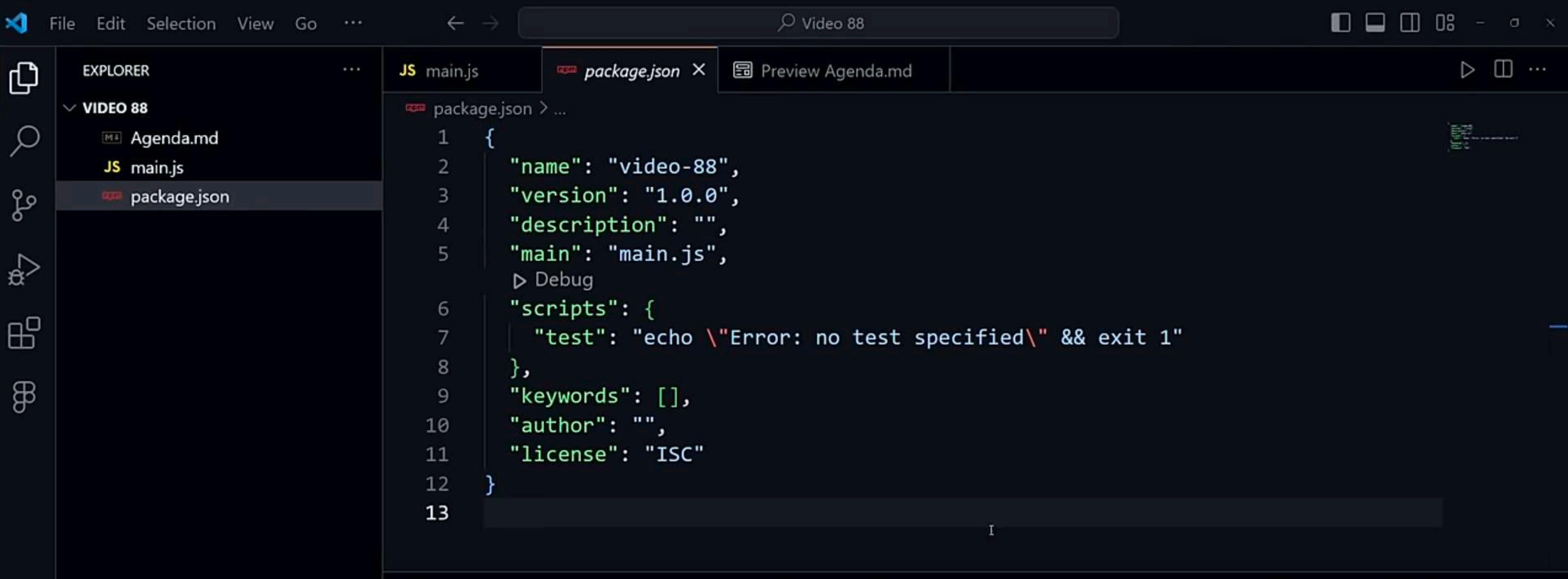
12°C Clear

Search

1932 Wed

Subscribe





File Edit Selection View Go ... ← → Video 88

EXPLORER ... JS main.js package.json X Preview Agenda.md

VIDEO 88

Agenda.md

JS main.js

package.json

```
package.json > ...
```

```
1  {
2   "name": "video-88",
3   "version": "1.0.0",
4   "description": "",
5   "main": "main.js",
6   ▷ Debug
7   "scripts": {
8     "test": "echo \"Error: no test specified\" && exit 1"
9   },
10  "keywords": [],
11  "author": "",
12  "license": "ISC"
13 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Code + - ×

```
Server running at http://127.0.0.1:3000/
[nodemon] restarting due to changes...
[nodemon] starting `node main.js`
Server running at http://127.0.0.1:3000/
PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 88> ^C
PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 88> ^C
PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 88> npm i express
[.....] - idealTree:Video 88: sill idealTree buildDeps
```

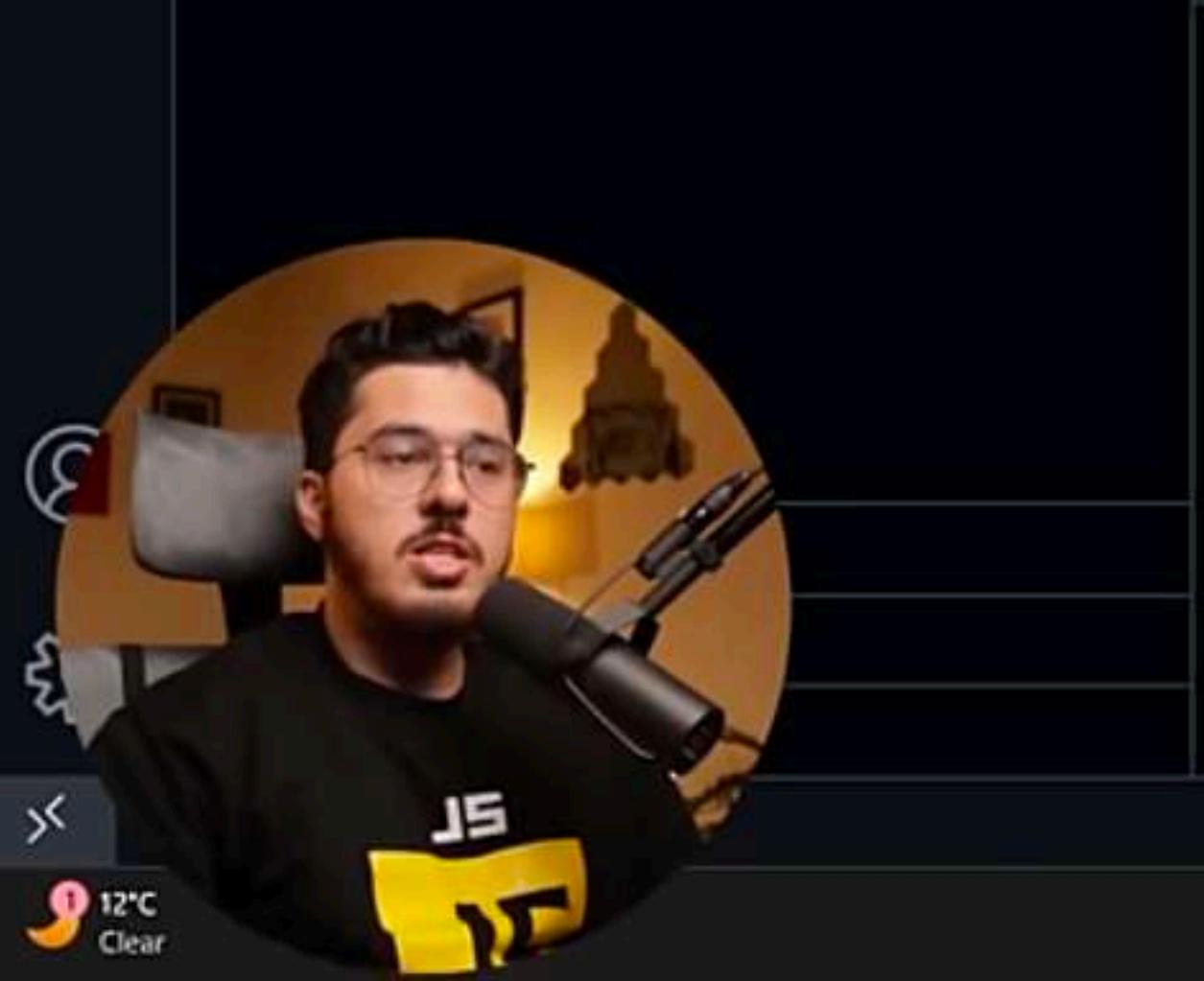
Ln 13, Col 1 Spaces: 2 UTF-8 LF {} JSON

12°C Clear

Search

1934 Wed

Subscribe



File Edit Selection View Go ... ← → Video 88

EXPLORER ... JS main.js package.json X Preview Agenda.md

VIDEO 88

node_modules Agenda.md main.js package-lock.json package.json

```
package.json > {} dependencies > express
  4   "description": "",
  5   "main": "main.js",
  6   ▶ Debug
  7   "scripts": {
  8     "test": "echo \"Error: no test specified\" && exit 1"
  9   },
 10   "keywords": [],
 11   "author": "",
 12   "license": "ISC",
 13   "dependencies": {
 14     "express": "^4.18.2"
 15   }
 16 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Code + ×

added 62 packages, and audited 63 packages in 3s

11 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities

PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 88> npm i express@4

Ln 13, Col 13 (12 selected) Spaces: 2 UTF-8 LF {} JSON

Subscribe

Sigma Web Development Course - W | Introduction to Node.js | Node.js | 127.0.0.1:3000 | Express "Hello World" example

expressjs.com/en/starter/hello-world.html

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C...

Black Lives Matter.
Support the Equal Justice Initiative.

Express

search Home Getting started Guide API reference Advanced topics Resources

Hello world example

Embedded below is essentially the simplest Express app you can create. It is a single file app — **not** what you'd get if you use the [Express generator](#), which creates the scaffolding for a full app with numerous JavaScript files, Jade templates, and sub-directories for various purposes.

```
1 const express = require('express' 4.18.2 )
2 const app = express()
3 const port = 3000
4
5 app.get('/', (req, res) => {
6   res.send('Hello World!')
7 })
8
9 app.listen(port, () => {
10   console.log(`Example app listening on port ${port}`)
11 })
```

Save on RunKit Node 10 ⚡ help URL: <https://slyu3xxsd6x.runkit.sh>

This app starts a server and listens on port 3000 for connections. The app responds with "Hello World!" for requests to the root URL (/) or **route**. For every other path, it will respond with a **404 Not Found**.

The example above is actually a working server: Go ahead and click on the URL shown. You'll get a response, with real-time logs on the page, and any changes you make will be reflected in real time. This is powered by [RunKit](#), which provides an interactive JavaScript playground connected to a complete Node environment that runs in your web browser. Below are instructions for running the same app on your local machine.

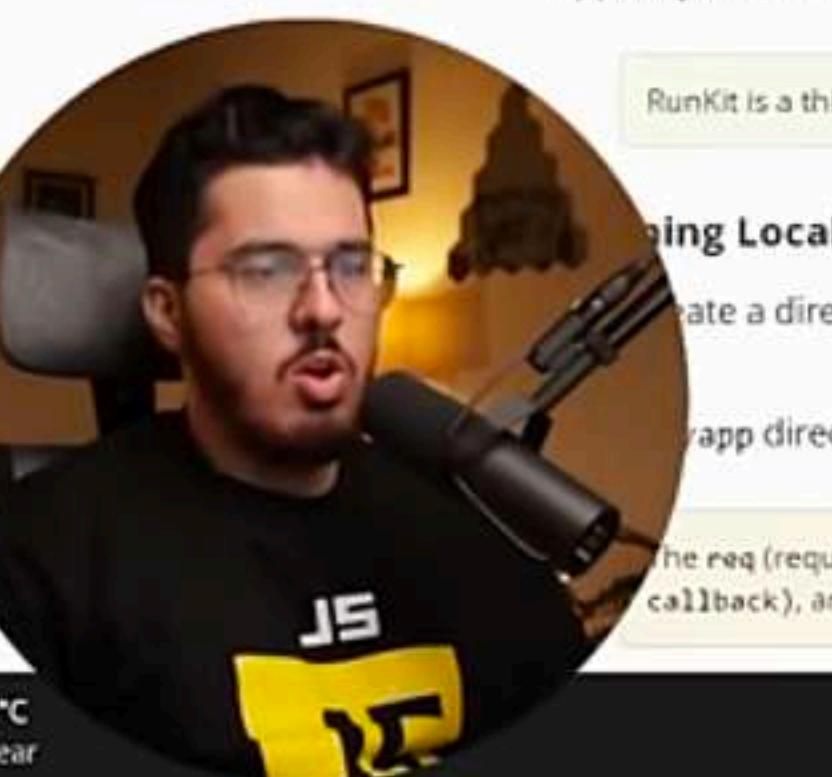
RunKit is a third-party service not affiliated with the Express project.

Running Locally

Create a directory named `myapp`, change to it and run `npm init`. Then install `express` as a dependency, as per the installation instructions.

In the `myapp` directory, create a file named `app.js` and copy in the code from the example above.

The `req` (request) and `res` (response) are the exact same objects that Node provides, so you can invoke `req.pipe()`, `req.on('data', callback)`, and anything else you would do without Express involved.

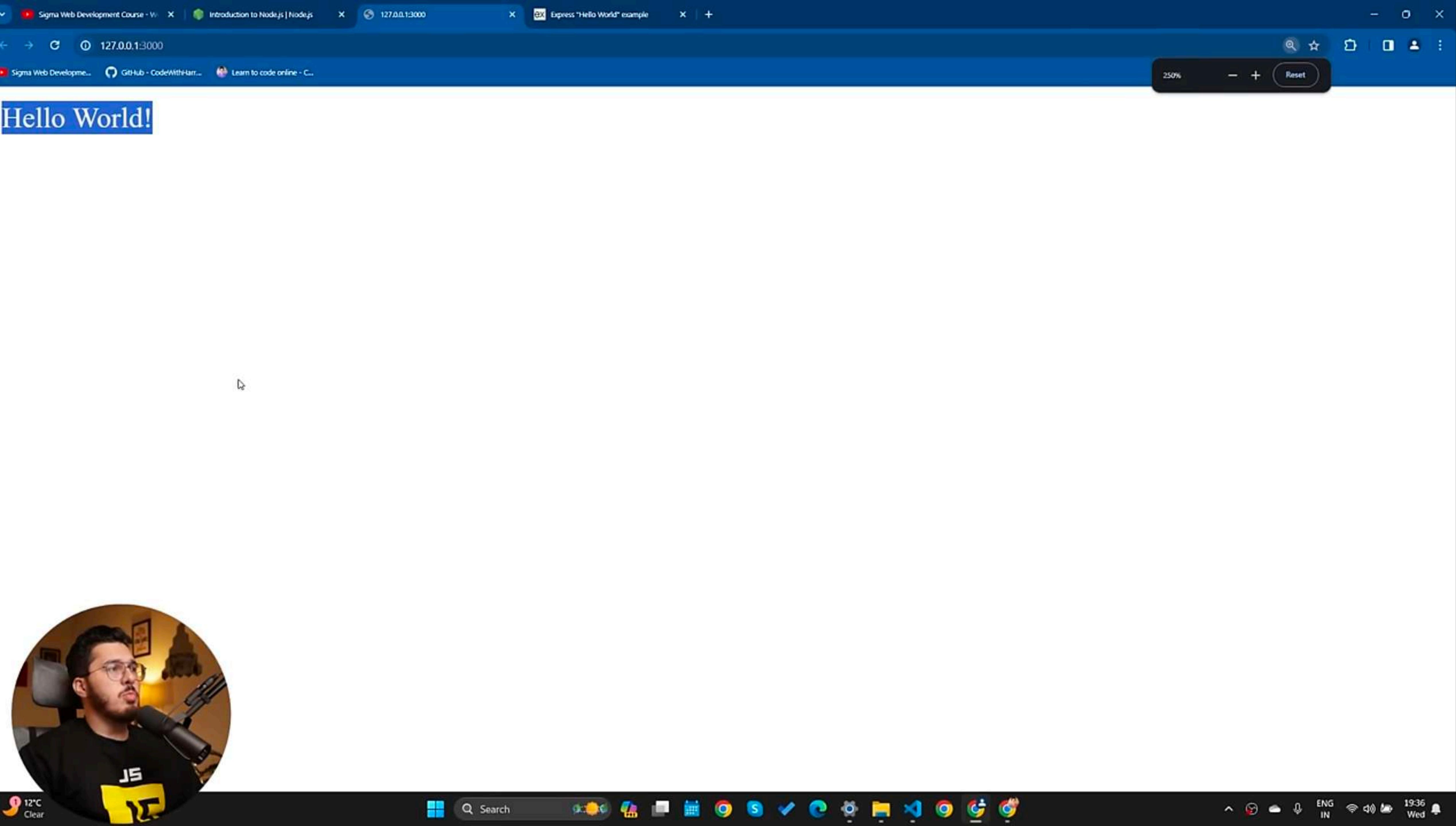


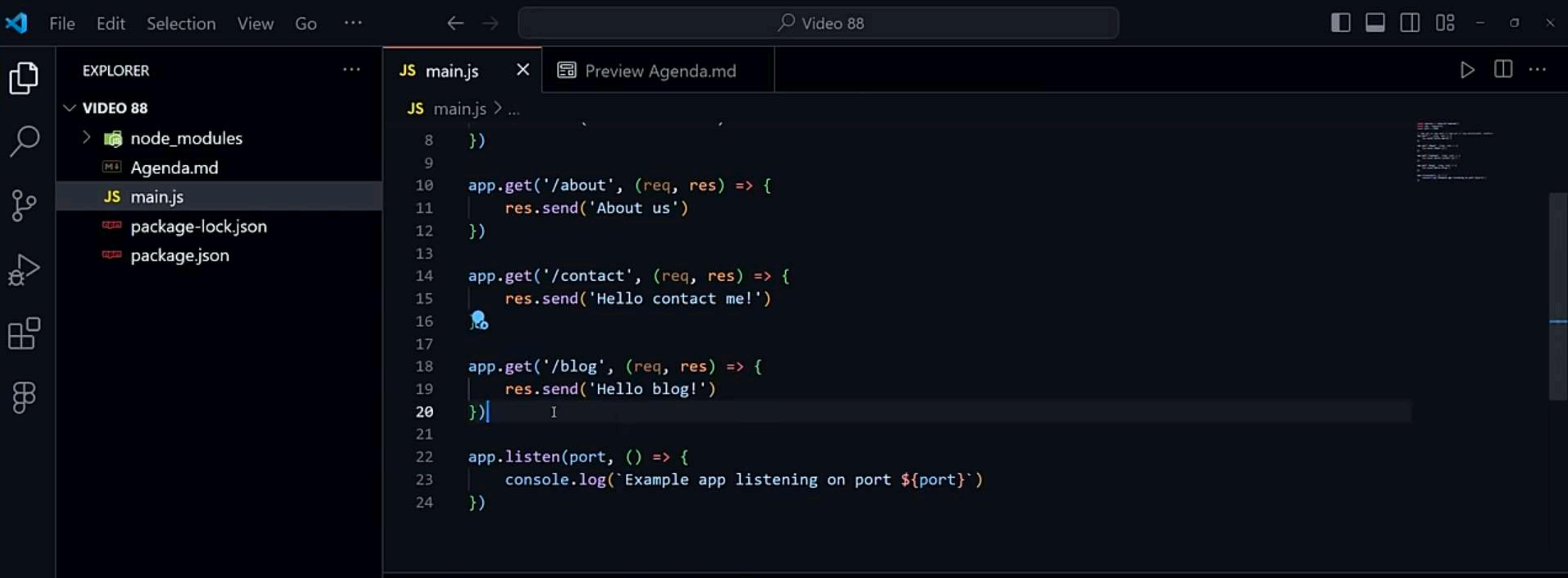
12°C Clear

Subscribe

Search

1936 Wed





File Edit Selection View Go ... ← → Video 88

EXPLORER

VIDEO 88

node_modules

Agenda.md

JS main.js

package-lock.json

package.json

JS main.js

JS main.js > ...

```
8  })
9
10 app.get('/about', (req, res) => {
11   res.send('About us')
12 })
13
14 app.get('/contact', (req, res) => {
15   res.send('Hello contact me!')
16 }
17
18 app.get('/blog', (req, res) => {
19   res.send('Hello blog!')
20 })
21
22 app.listen(port, () => {
23   console.log(`Example app listening on port ${port}`)
24 })
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

[nodemon] restarting due to changes...

[nodemon] starting `node main.js`

Example app listening on port 3000

[nodemon] restarting due to changes...

[nodemon] restarting due to changes...

[nodemon] starting `node main.js`

Example app listening on port 3000

Ln 20, Col 3 Spaces: 4 UTF-8 CRLF {} JavaScript

12°C Clear

Search

19:40 Wed

Subscribe

File Edit Selection View Go ... ← → ⚡ Video 88

EXPLORER ... JS main.js X Preview Agenda.md

VIDEO 88 > node_modules Agenda.md JS main.js package-lock.json package.json

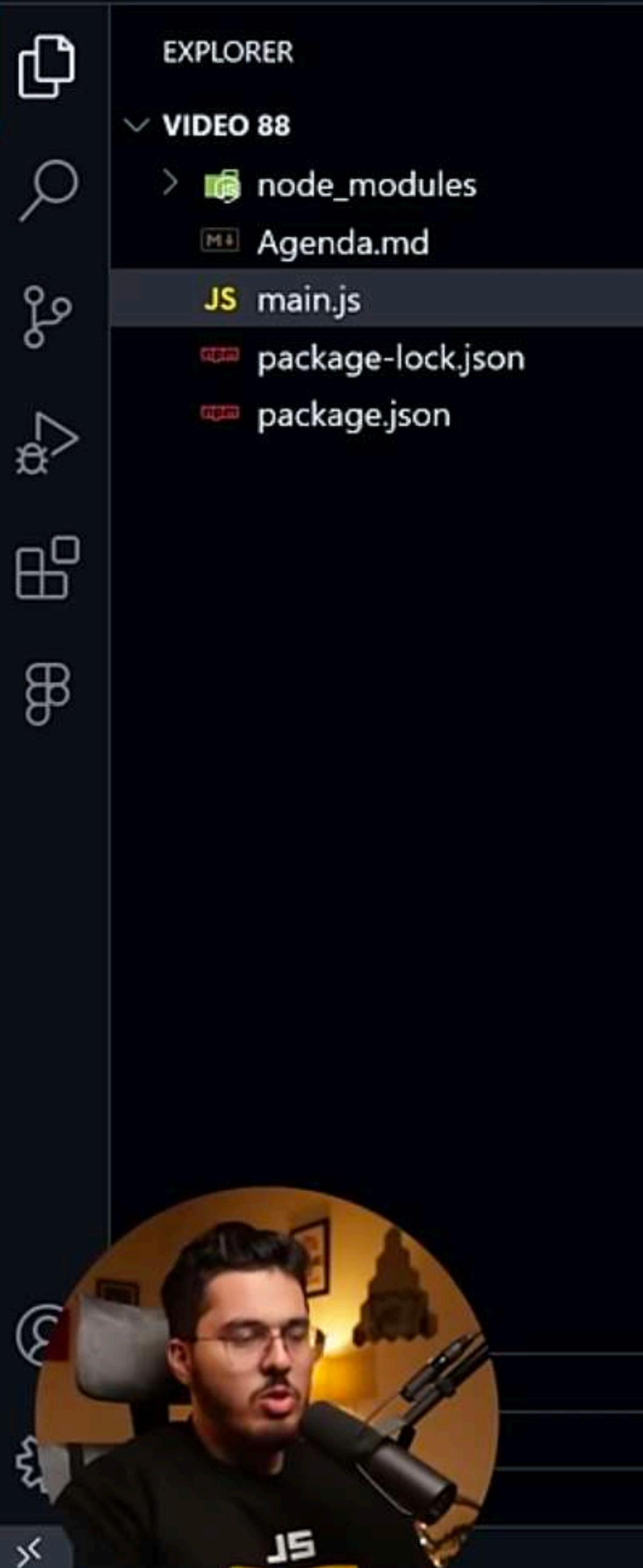
JS main.js > ...

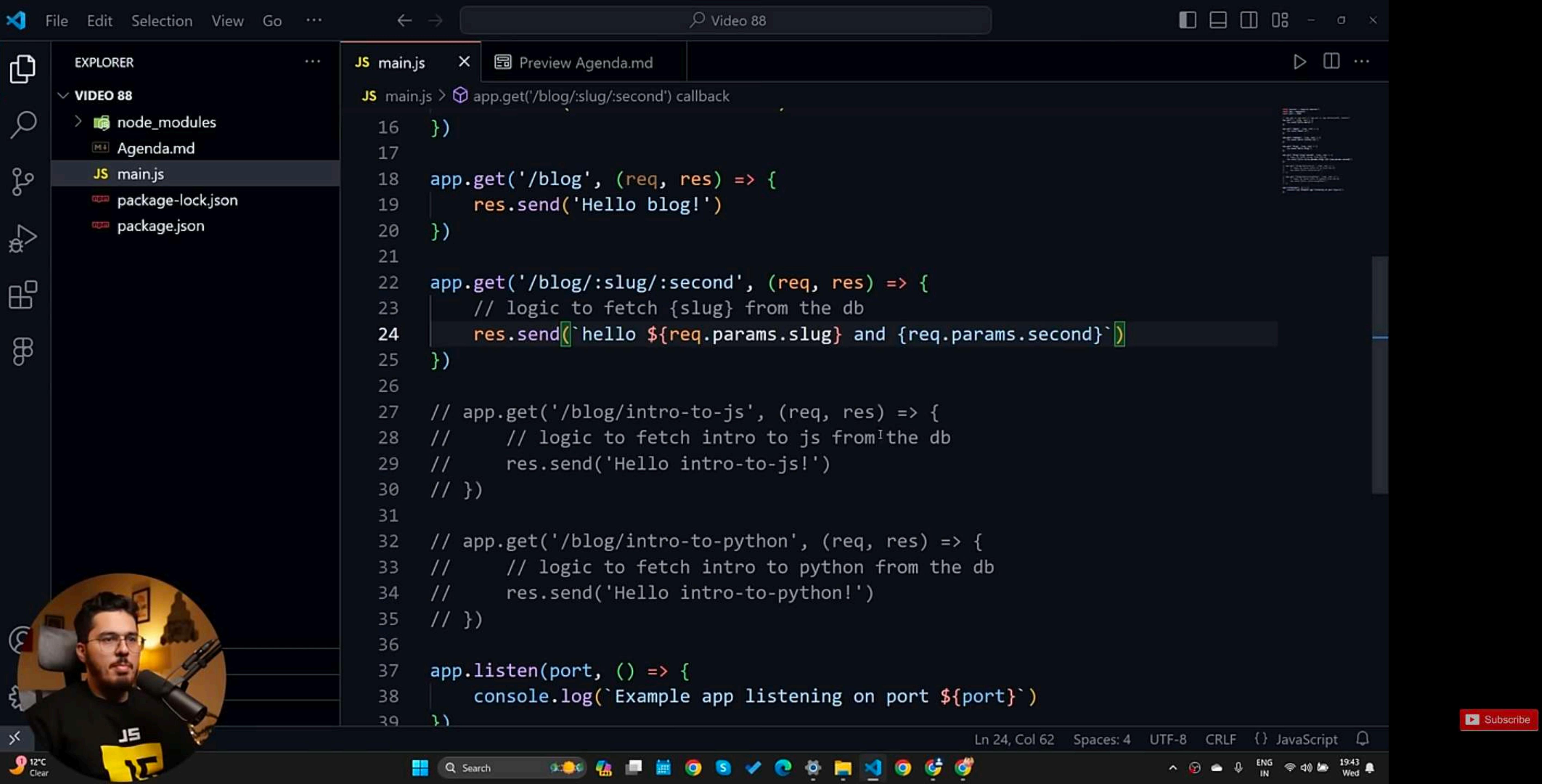
```
16  })
17
18  app.get('/blog', (req, res) => {
19      res.send('Hello blog!')
20
21  app.get('/blog/intro-to-js', (req, res) => {
22      // logic to fetch intro to js from the db
23      res.send('Hello intro-to-js!')
24
25  })
26
27  app.get('/blog/intro-to-python', (req, res) => {
28      // logic to fetch intro to python from the db
29      res.send('Hello intro-to-python!')
30
31
32  app.listen(port, () => {
33      console.log(`Example app listening on port ${port}`)
34  })
```

Ln 21, Col 1 (280 selected) Spaces: 4 UTF-8 CRLF {} JavaScript

12°C Clear

Subscribe







File Edit Selection View Go ... ← → Video 88

EXPLORER ... JS main.js X Preview Agenda.md

VIDEO 88

node_modules

Agenda.md

JS main.js

package-lock.json

package.json

```
JS main.js > app.get('/blog/:slug') callback
22 app.get('/blog/:slug', (req, res) => {
23   // logic to fetch {slug}
24   // For URL: http://localhost:3000/blog/intro-to-padosi?mode=dark&region=in
25   console.log(req.params) // will output { slug: 'intro-to-padosi' }
26   console.log(req.query) // will output { mode: 'dark', region: 'in' }
27
28   res.send(`hello ${req.params.slug}`)
29 }
30
31 // app.get('/blog/intro-to-js', (req, res) => {
32 //   // logic to fetch intro to js from the db
33 //   res.send('Hello intro-to-js!')
34 // }
35
36 // app.get('/blog/intro-to-python', (req, res) => {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Code + ×

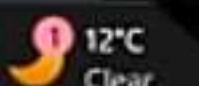
PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 88> nodemon main.js

```
[nodemon] 3.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node main.js`
Example app listening on port 3000
{ slug: 'intro-to-padosi' }
{ mode: 'dark', region: 'in' }
[nodemon] restarting due to changes...
[nodemon] starting `node main.js`
Example app listening on port 3000
```

⋮

Ln 28, Col 41 Spaces: 4 UTF-8 CRLF {} JavaScript

Subscribe



Sigma Web Development Course - W | Introduction to Node.js | Node.js | Express "Hello World" example | localhost:3000/blog/intro-to-padosi | Google Search

expressjs.com/en/starter/hello-world.html

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C...

Black Lives Matter. Support the Equal Justice Initiative.

Express

search Home Getting started Guide API reference Advanced topics Resources

Hello world example

Embedded below is essentially the simplest Express generator, which creates the scaffolding for a file app — **not** what you'd get if you use the Express generator, Jade templates, and sub-directories for various purposes.

```
1 const express = require('express') 4: 2 const app = express() 3 const port = 3000 4 5 app.get('/', (req, res) => { 6   res.send('Hello World!') 7 }) 8 9 app.listen(port, () => { 10   console.log(`Example app listening on port ${port}`) 11 })
```

Save on RunKit Node 10 ⚡ help URL: <https://slyu3xxsd6x.runkit.sh>

This app starts a server and listens on port 3000 for connections. The app responds with "Hello World!" for requests to the root URL (/) or **route**. For every other path, it will respond with a **404 Not Found**.

The example above is actually a working server: Go ahead and click on the URL shown. You'll get a response, with real-time logs on the page, and any changes you make will be reflected in real time. This is powered by RunKit, which provides an interactive JavaScript playground connected to a complete Node environment that runs in your web browser. Below are instructions for running the same app on your local machine.

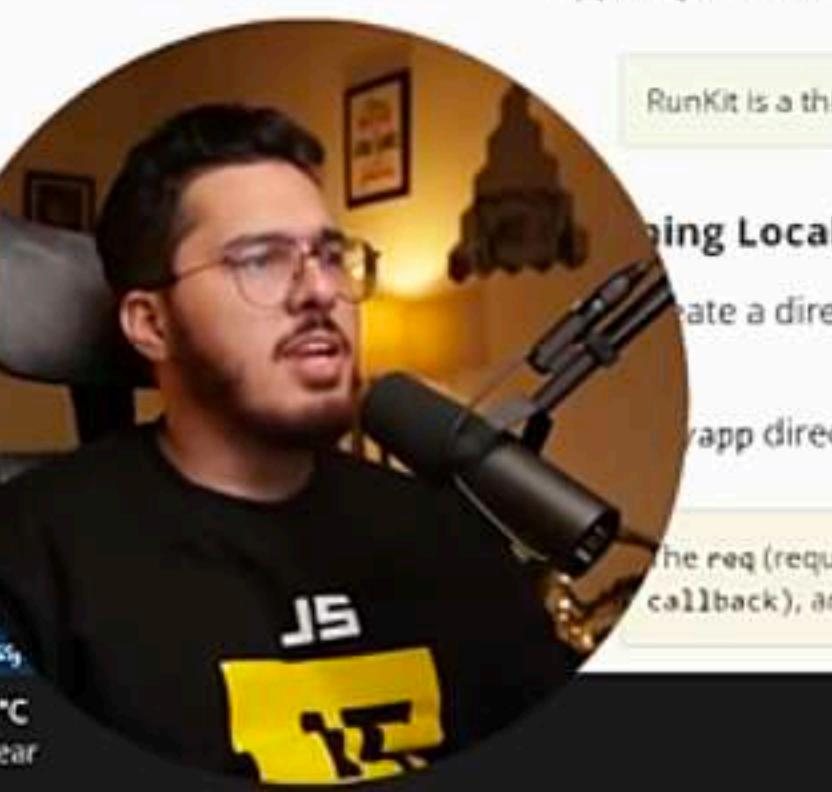
RunKit is a third-party service not affiliated with the Express project.

Running Locally

Create a directory named `myapp`, change to it and run `npm init`. Then install `express` as a dependency, as per the installation instructions.

In the `myapp` directory, create a file named `app.js` and copy in the code from the example above.

The `req` (request) and `res` (response) are the exact same objects that Node provides, so you can invoke `req.pipe()`, `req.on('data', callback)`, and anything else you would do without Express involved.



Subscribe

12°C Clear

Search

ENG IN

19:52 Wed

Sigma Web Development Course - V | Introduction to Node.js | Node.js | Serving static files in Express | localhost3000/blog/intro-to-padosh | hey - Google Search

expressjs.com/en/starter/static-files.html

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C...

Black Lives Matter.
Support the Equal Justice Initiative.

Express

search Home Getting started Guide API reference Advanced topics Resources

Serving static files in Express

To serve static files such as images, CSS files, and JavaScript files, use the `express.static` built-in middleware function in Express.

The function signature is:

```
express.static(root, [options])
```

The `root` argument specifies the root directory from which to serve static assets. For more information on the `options` argument, see [express.static](#).

For example, use the following code to serve images, CSS files, and JavaScript files in a directory named `public`:

```
app.use(express.static('public'))
```

Now, you can load the files that are in the `public` directory:

```
http://localhost:3000/images/kitten.jpg  
http://localhost:3000/css/style.css  
http://localhost:3000/js/app.js  
http://localhost:3000/images/bg.png  
http://localhost:3000/hello.html
```

Express looks up the files relative to the static directory, so the name of the static directory is not part of the URL.

To use multiple static assets directories, call the `express.static` middleware function multiple times:

```
app.use(express.static('public'))  
app.use(express.static('files'))
```

Express looks up the files in the order in which you set the static directories with the `express.static` middleware function.

NOTE: For best results, use a reverse proxy cache to improve performance of serving static assets.

Create a virtual path prefix (where the path does not actually exist in the file system) for files that are served by the `express.static` function, specify a mount path for the static directory, as shown below:

```
app.use('/static', express.static('public'))
```



12°C Clear

Search Google S E C G

ENG IN 19:52 Wed

Subscribe



Response, Request and Routers in Express | Sigma Web Development Course - Tutorial...

139K views 1y ago Sigma Web Development Course - Web ...more



CodeWithHarry 8.3M

Join



4.1K | 139K Share Remix Download

Comments 376



Going to name my village land to yours...but the problem is... I even don't have access to my Village land... 😂😂



Multi-Tenant Authentication for B2B SaaS

End of playlist

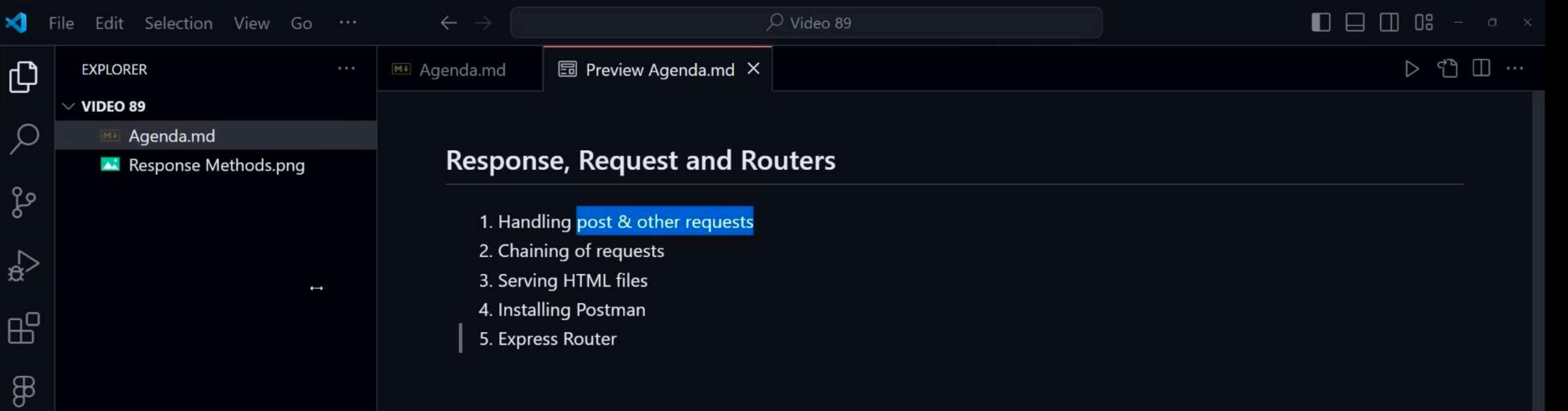


Today's Special • 4/4

Walkthrough how to build a complete multi-tenant

experience into your app

Sponsored : Clerk



File Edit Selection View Go ... ← → Video 89 □ □ □ 0% - □ ×

EXPLORER ...

VIDEO 89

Agenda.md

Preview Agenda.md X

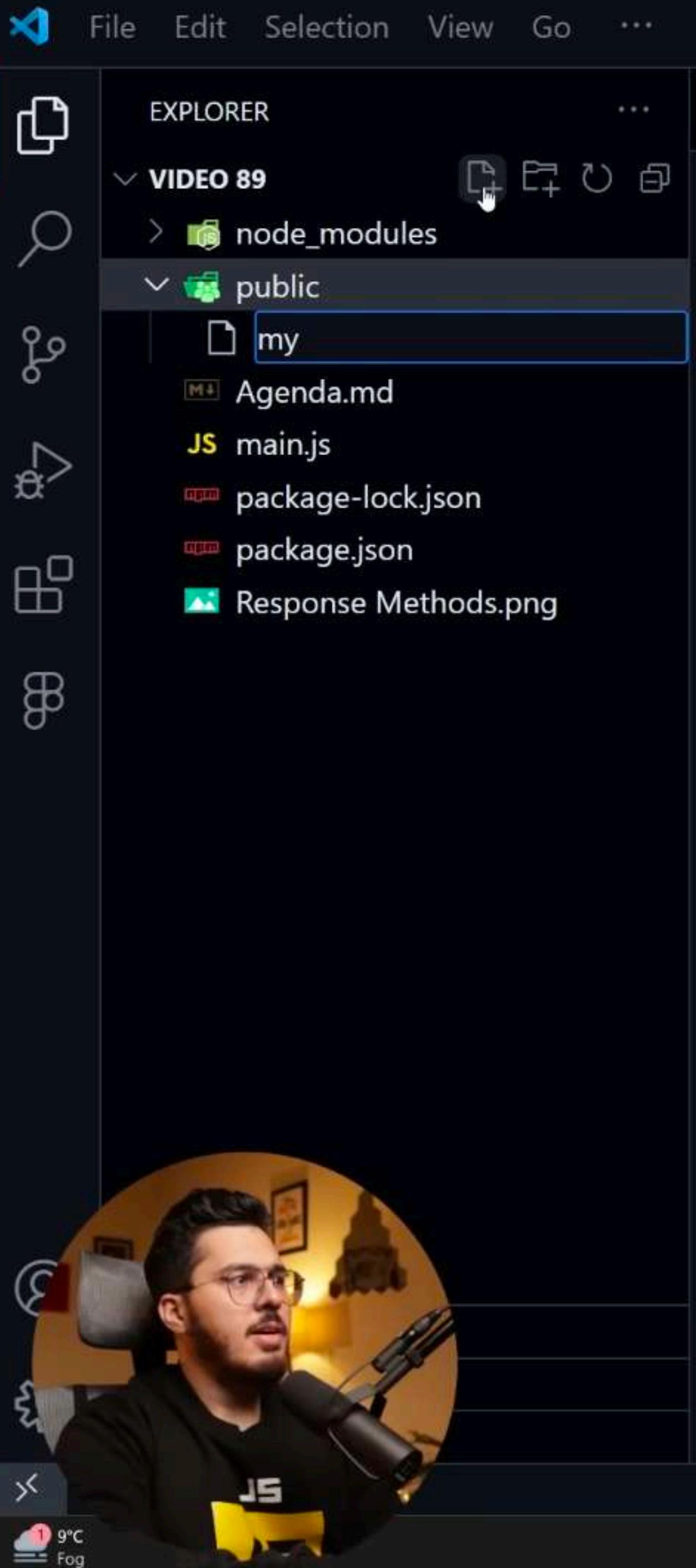
Agenda.md

Response Methods.png

Response, Request and Routers

1. Handling post & other requests
2. Chaining of requests
3. Serving HTML files
4. Installing Postman
5. Express Router





File Edit Selection View Go ... ← → Video 89 □ □ □ □ - □ ×

EXPLORER ...

VIDEO 89

node_modules

public

my

Agenda.md

main.js

package-lock.json

package.json

Response Methods.png

Preview Agenda.md JS main.js X Code

JS main.js > ...

```
1 const express = require('express')
2 const app = express()
3 const port = 3000
4
5 app.use(express.static("public"))
6
7 app.get('/', (req, res) => {
8   console.log("Hey its a get request")
9   res.send('Hello World2!')
10 })
11
12 app.post('/', (req, res) => {
13   console.log("Hey its a post request")
14   res.send('Hello World post!')
15 })
16
17 app.listen(port, () => {
18   console.log(`Example app listening on port ${port}`)
19 })
```

Ln 5, Col 31 Spaces: 4 UTF-8 CRLF {} JavaScript

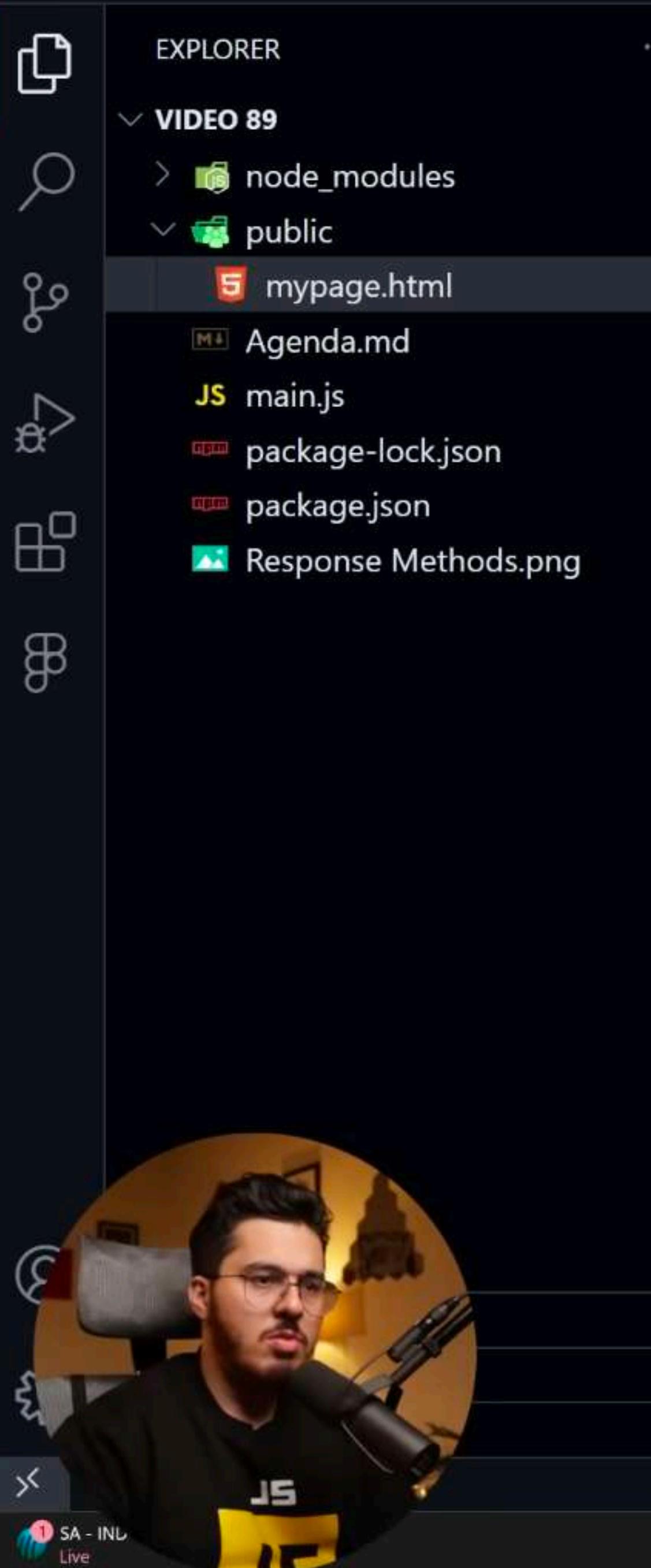
19°C Fog

Search

02:04 Thu

ENG IN

Subscribe



File Edit Selection View Go ... ← → Video 89

EXPLORER ...

VIDEO 89

- > node_modules
- public
- mypage.html
- Agenda.md
- main.js
- package-lock.json
- package.json
- Response Methods.png

Preview Agenda.md main.js mypage.html X Code

```
public > 5 mypage.html > html > body > script > testPost > a
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width,
6     initial-scale=1.0">
6   <title>Testing Post Request</title>
7 </head>
8 <body>
9   <script>
10     async function testPost(){
11       let a = await fetch("/", {method: "POST"})
12       let b = await a.text()
13       console.log(b)
14     }
15
16     testPost()
17   </script>
18 </body>
19 </html>
```

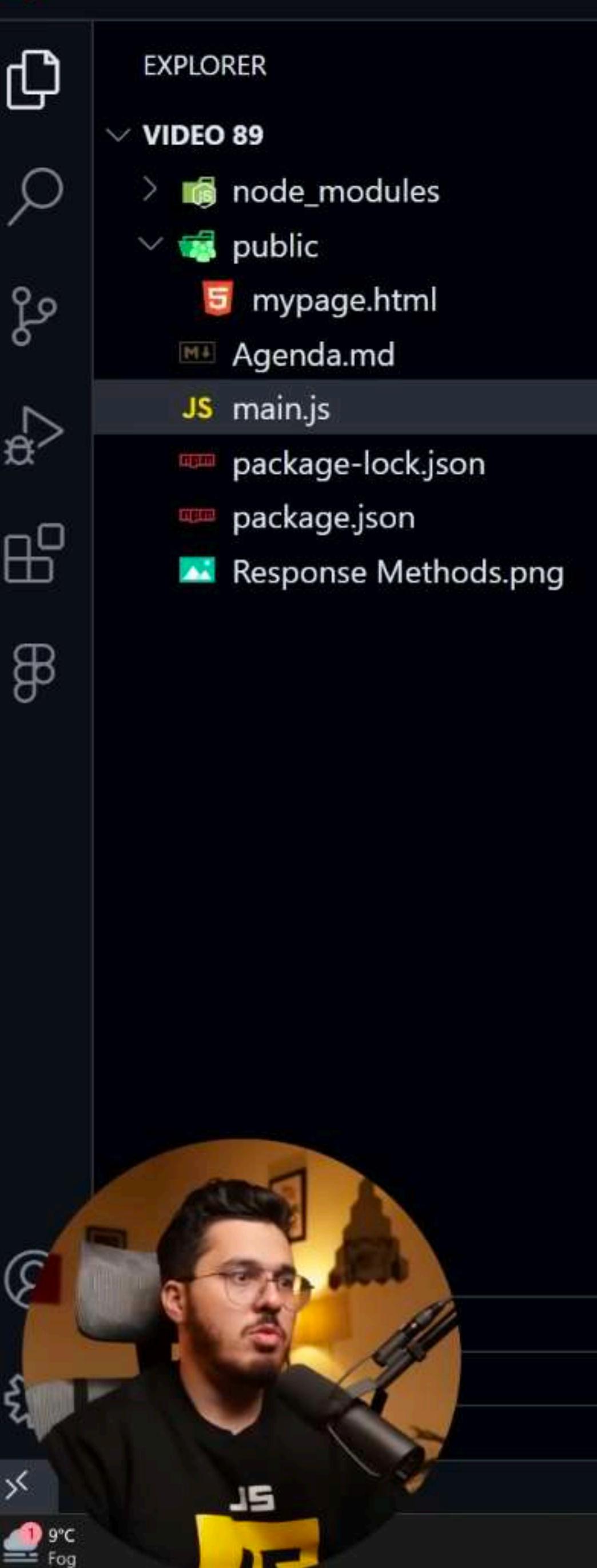
Ln 11, Col 35 Spaces: 4 UTF-8 CRLF HTML

1 SA - IND Live

Search

02:06 Thu

Subscribe



File Edit Selection View Go ... ← → Video 89 □ □ □ □ - □ ×

EXPLORER ...

VIDEO 89

- > node_modules
- public
- mypage.html
- Agenda.md
- main.js
- package-lock.json
- package.json
- Response Methods.png

Preview Agenda.md JS main.js X mypage.html Code

```
JS main.js > ...
3 const port = 3000
4
5 app.use(express.static("public"))
6
7 app.get('/', (req, res) => {
8     console.log("Hey its a get request")
9     res.send('Hello World2!')
10})
11
12 app.post('/', (req, res) => {
13     console.log("Hey its a post request")
14     res.send('Hello World post!')
15})
16
17 app.put('/', (req, res) => {
18     console.log("Hey its a put request")
19     res.send('Hello World put!')
20})
21
22 app.listen(port, () => {
23     console.log(`Example app listening on port ${port}`)
24})
```

Ln 11, Col 1 (107 selected) Spaces: 4 UTF-8 CRLF {} JavaScript

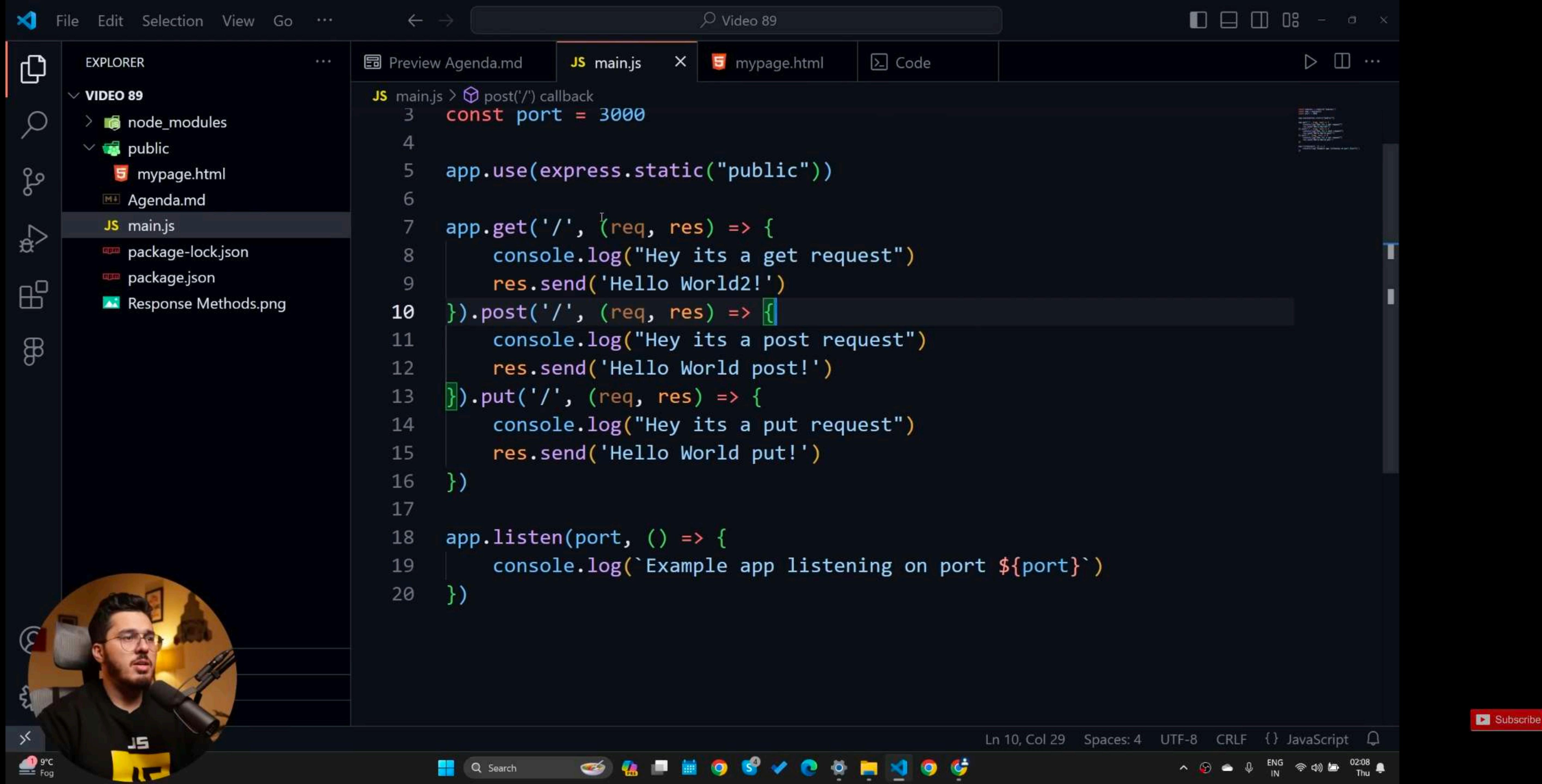
19°C Fog

Search

02:08 Thu

ENG IN

Subscribe



EXPLORER ...

VIDEO 89

> node_modules

> public

templates

index.html

Agenda.md

main.js

package-lock.json

package.json

Response Methods.png

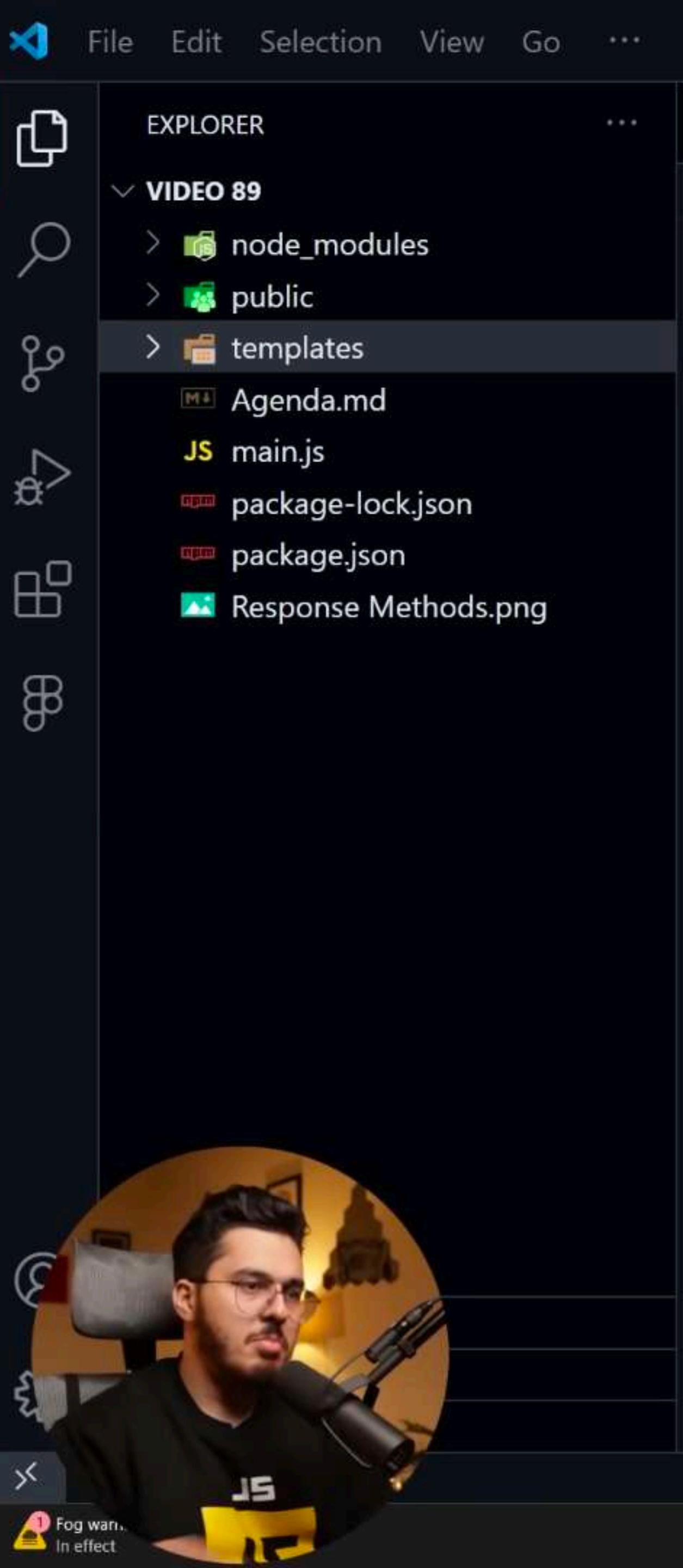
Preview Agenda.md JS main.js index.html X Code

templates > index.html > html > body > b

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.
7     0">
8   <title>Document</title>
9 </head>
10 <body>
11   <b>Hey I am bold</b>
12   This is index.html
13 </body>
14
15 </html>
```



Subscribe



File Edit Selection View Go ... ← → Video 89 □ □ □ □ - □ ×

EXPLORER ...

VIDEO 89

- > node_modules
- > public
- > templates
- Agenda.md
- main.js
- package-lock.json
- package.json
- Response Methods.png

Preview Agenda.md JS main.js X index.html Code

```
JS main.js > app.get("/index") callback
  8   console.log("Hey its a get request")
  9   res.send('Hello World2!')
10  }).post('/', (req, res) => {
11    console.log("Hey its a post request")
12    res.send('Hello World post!')
13  )
14
15  app.put('/', (req, res) => {
16    console.log("Hey its a put request")
17    res.send('Hello World put!')
18  )
19
20  app.get("/index", (req, res) => {
21    console.log("Hey its index")
22    res.sendFile('templates/index.html', {root: __dirname})
23  )
24
25  app.listen(port, () => {
26    console.log(`Example app listening on port ${port}`)
27  )
```

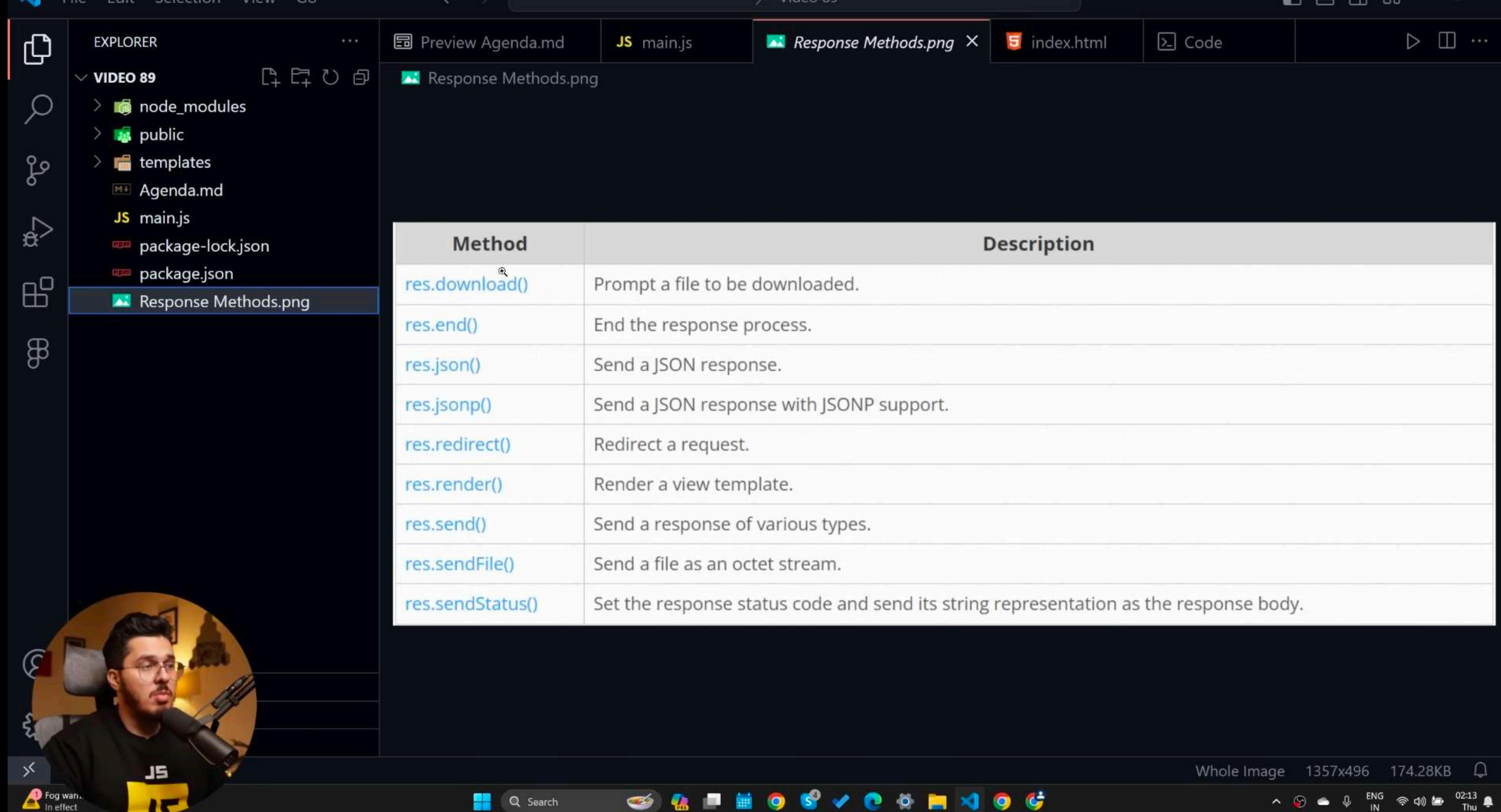
Ln 22, Col 24 Spaces: 4 UTF-8 CRLF {} JavaScript

Subscribe

Fog warn: In effect

Search

02:12 Thu



Postman API Platform

Download Postman | Get Started for

localhost:3000/api

postman.com/downloads/

CodeWithHarry - YouTube Tailwind CSS Tutorials in... Next.js Tutorials for Begi... Read-toastify | React-To... Strapi Tutorial For Beginner... GitHub Sanity.io Tutorials in Hin... JavaScript Tutorials for Begi... JavaScript Tutorial for Begi... GUESS THE NUMBER ... Exercise No 1 - Nix (beta... Js Notes New Tab

Product Pricing Enterprise Resources and Support Public API Network

Contact Sales Launch Postman

Download Postman

Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman.

The Postman app

Download the app to get started with the Postman API Platform.

[Windows 64-bit](#)

By downloading and using Postman, I agree to the [Privacy Policy](#) and [Terms](#).

[Release Notes](#)

Not your OS? Download for Mac ([Intel Chip](#), [Apple Chip](#)) or Linux ([x64](#), [arm64](#))



Postman on the web

Access the Postman API Platform through your web browser. Create a free account, and you're in.

[Launch Postman](#)

YOU CAN ALSO TRY THE WEB VERSION OF POSTMAN AT [WEB.POSTMAN.CO](https://web.postman.co)

https://dl.pstn...

1°C Fog

109 110

Manage Cookies Reject All Cookies Accept All Cookies

02:17 Thu

Subcribe

Home Workspaces API Network Explore

Notion's Public Workspace New Import Overview

GET Retrieve a database

... / Databases / Retrieve a database

Save Send

Documentation

https://api.notion.com/v

Retrieves a database obje in the request path.

Authorization Bearer

This request is using an au collection Notion API

Request Header

Notion-Version 22-0

Path Variables

id

Body

Prett

We use cookies on your device to enhance your navigation experience, analyze usage to improve our site, and customize our marketing efforts. You can learn more about cookies in our [Cookie Notice](#).

EXPLORER ...

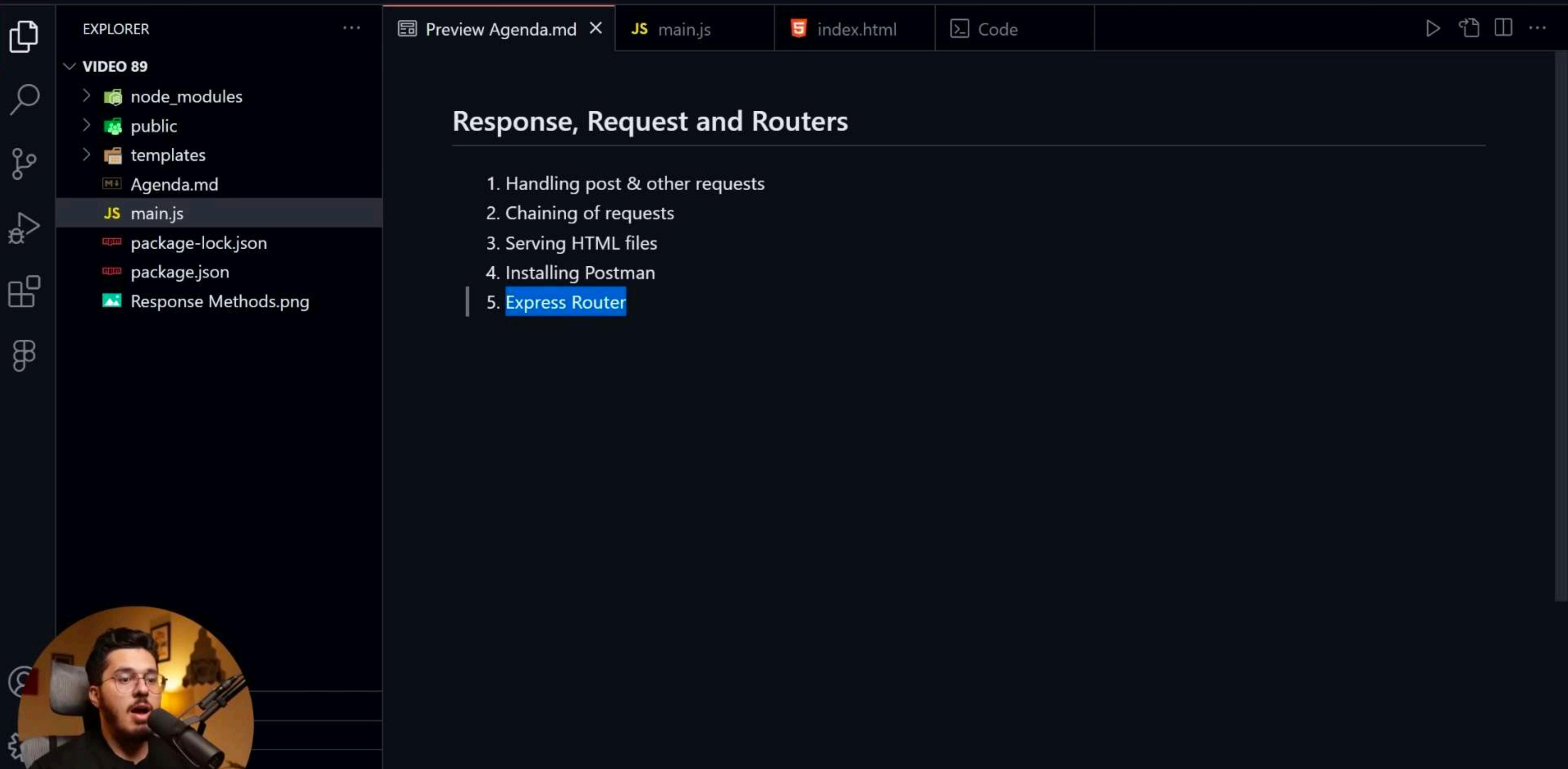
VIDEO 89

- > node_modules
- > public
- > templates
- Agenda.md
- main.js**
- package-lock.json
- package.json
- Response Methods.png

Preview Agenda.md X JS main.js index.html Code

Response, Request and Routers

1. Handling post & other requests
2. Chaining of requests
3. Serving HTML files
4. Installing Postman
5. Express Router



EXPLORER ...

VIDEO 89

- > node_modules
- > public
- > routes
 - JS blog.js
 - > templates
 - Agenda.md
 - JS main.js
 - package-lock.json
 - package.json
 - Response Methods.png

routes > JS blog.js > router.get('/about') callback

```
1 const express = require('express')
2 const router = express.Router()
3
4 // define the home page route
5 router.get('/', (req, res) => {
6   res.send('Blog home page')
7 })
8
9 // define the about route
10 router.get('/about', (req, res) => {
11   res.send('About blog')
12 })
13
14 // define the about route
15 router.get('/blogpost/:slug', (req, res) => {
16   res.send(`fetch the blogpost for ${req.params.slug}`)
17 })
18
19 module.exports = router
```





File Edit Selection View Go ... ← → Video 89 □ □ □ □ - □ ×

EXPLORER ...

VIDEO 89

- > node_modules
- > public
- > routes
 - JS blog.js
- > templates
- Agenda.md
- JS main.js
- package-lock.json
- package.json
- Response Methods.png

Preview Agenda.md JS main.js X JS blog.js JS index.html Code

JS main.js > ...

```
1 const express = require('express')
2 const blog = require('./routes/blog')
3
4
5
6 const app = express()
7 const port = 3000
8
9 app.use(express.static("public"))
10 app.use('/blog', blog)
11
12 app.get('/', (req, res) => {
13   console.log("Hey its a get request")
14   res.send('Hello World21!')
15 }).post('/', (req, res) => {
16   console.log("Hey its a post request")
17   res.send('Hello World post!')
18 })
19
20 app.put('/', (req, res) => {
21   console.log("Hey its a put request")
22   res.send('Hello World put!')
```

Ln 9, Col 33 Spaces: 4 UTF-8 CRLF {} JavaScript

19°C Fog

Search

02:28 Thu

Subscribe



Become A Python pro in...

Sponsored · aifortechies.in

Visit site

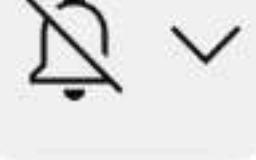
Middlewares in Express Js | Sigma Web Development Course - Tutorial #90

114K views 1y ago Sigma Web Development Course - We... [more](#)

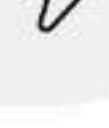


CodeWithHarry 8.34M

Join



4.5K



Share

Remix



Download

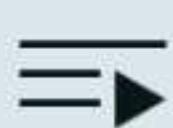
Comments 375



Vote for JSON Tutorial

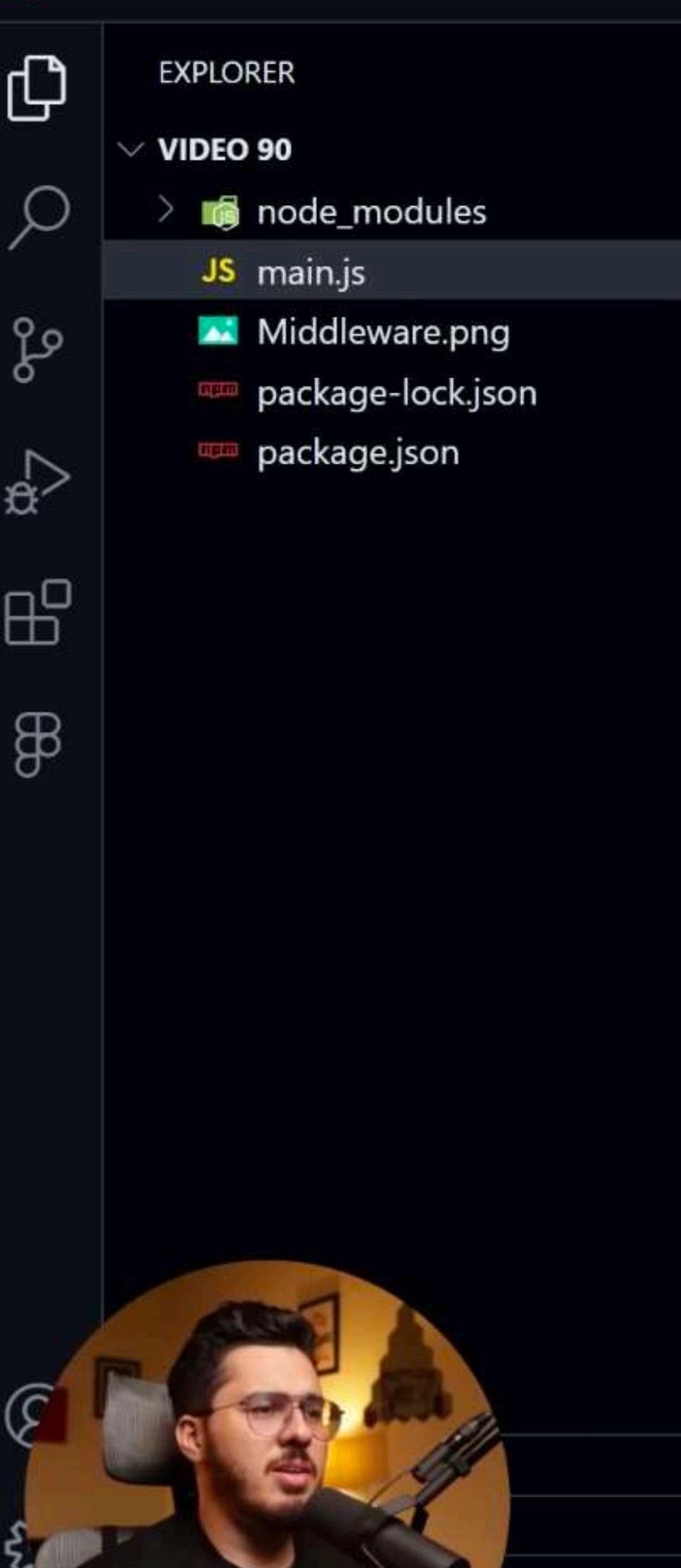


End of playlist



Today's Special • 4/4

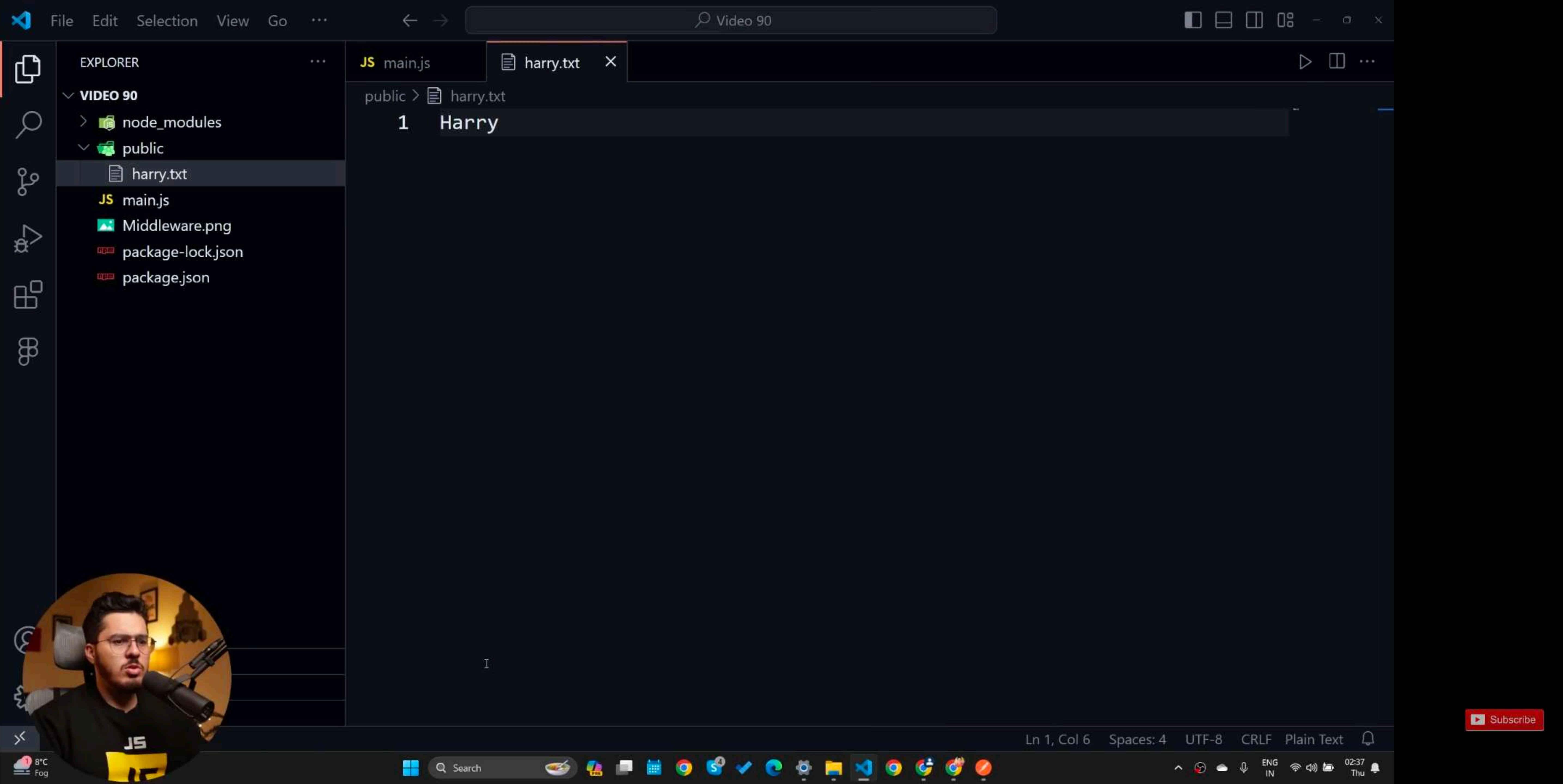


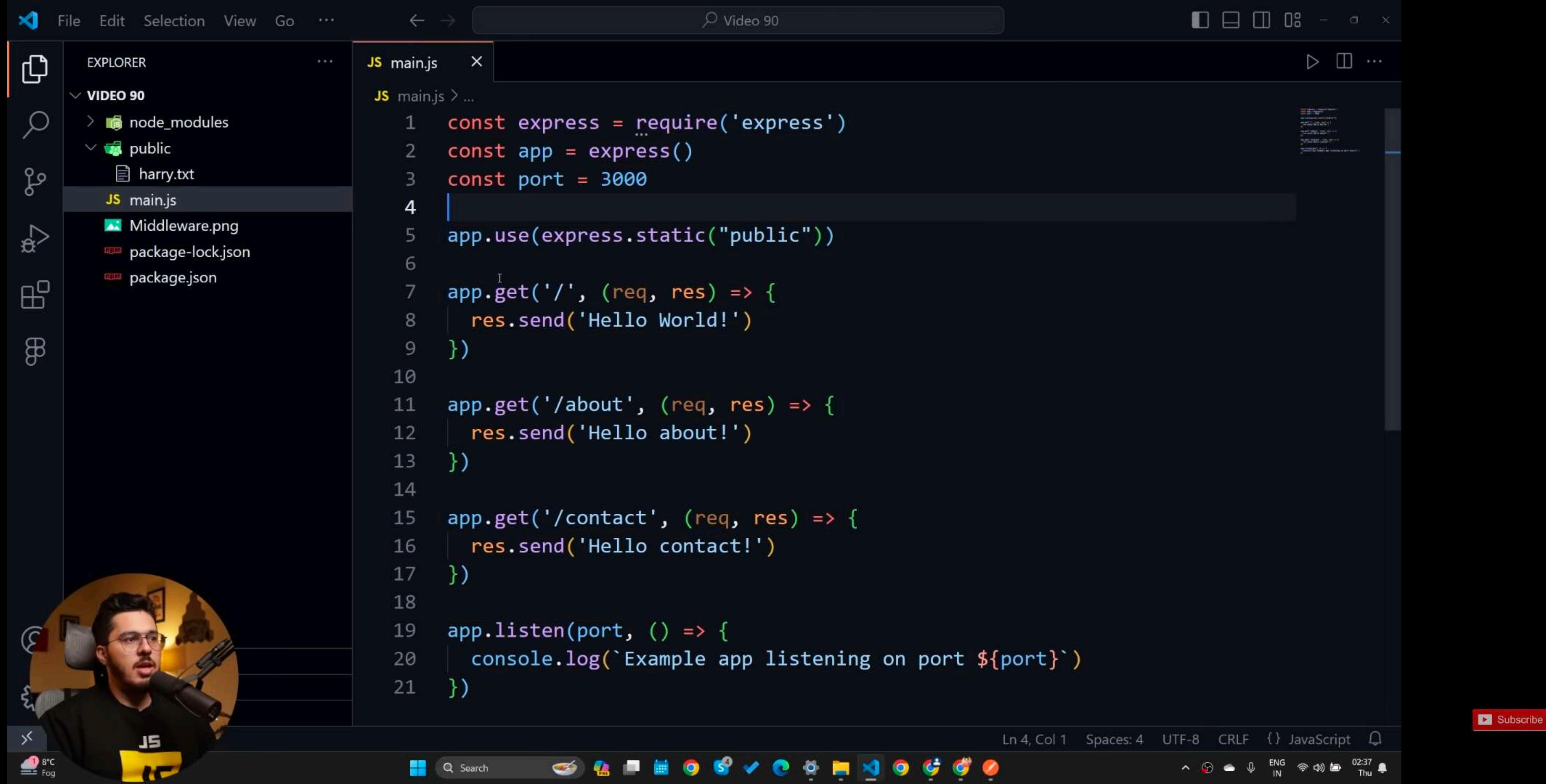


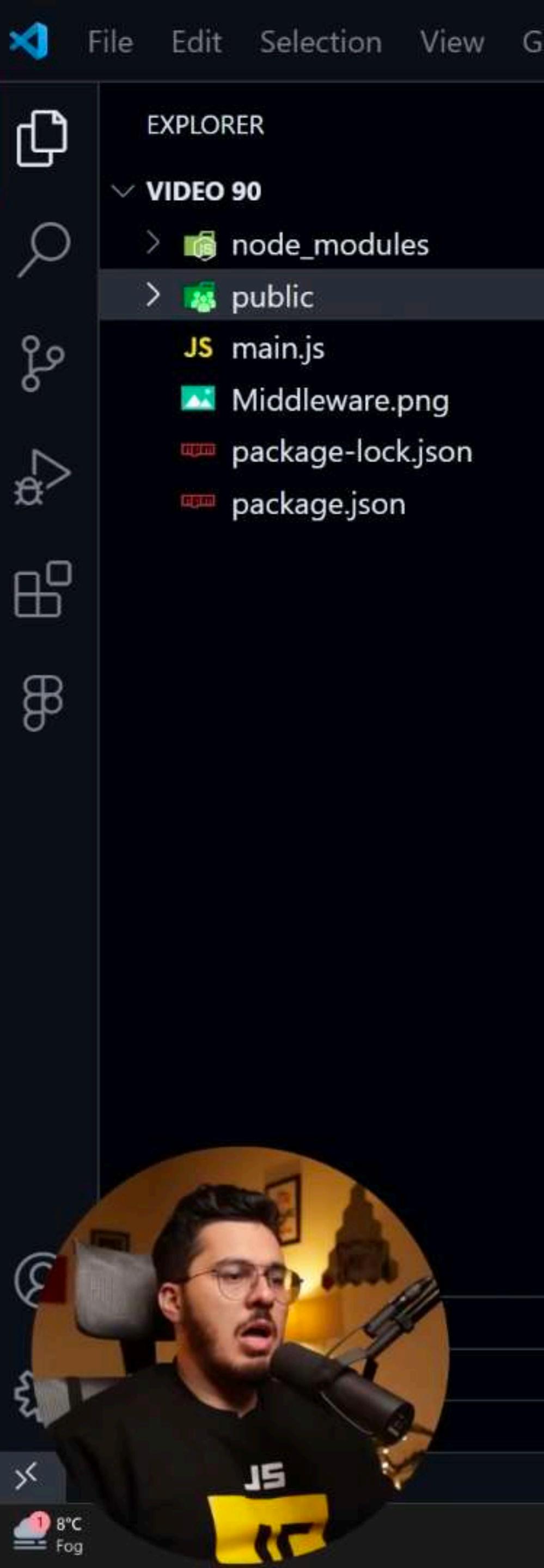
JS main.js X

JS main.js > ⚙ app.get('/about') callback

```
1  const express = require('express')
2  const app = express()
3  const port = 3000
4
5  app.get('/', (req, res) => {
6    |  res.send('Hello World!')
7  })
8
9  app.get('/about', (req, res) => {
10   |  res.send('Hello about!')
11 })
12
13 app.get('/contact', (req, res) => {
14   res.send('Hello contact!')
15 })
16
17 app.listen(port, () => {
18   |  console.log(`Example app listening on port ${port}`)
19 })
```







File Edit Selection View Go ... ← → 🔍 Video 90

EXPLORER ...

VIDEO 90

- > node_modules
- > public
- JS main.js
- Middleware.png
- package-lock.json
- package.json

JS main.js X

JS main.js > app.use() callback

```
1 const express = require('express')
2 const app = express()
3 const port = 3000
4
5 // app.use(express.static("public"))
6
7 // Middleware 1
8 app.use((req, res, next) => {
9   console.log('m1')
10  next()
11 })
12
13 // Middleware 2
14 app.use((req, res, next) => {
15   console.log('m2')
16   next()
17 })
18
19 app.get('/', (req, res) => {
20   res.send('Hello World!')
21 })
22
```

Ln 15, Col 20 Spaces: 4 UTF-8 CRLF {} JavaScript

18°C Fog

Subscribe

Windows Search File Explorer Task View Taskbar

File Edit Selection View Go ... ← → Video 90

EXPLORER ... JS main.js X

VIDEO 90 node_modules public main.js Middleware.png package-lock.json package.json

```
JS main.js > app.use() callback
4
5 // app.use(express.static("public"))
6
7 // Middleware 1
8 app.use((req, res, next) => {
9   console.log('m1')
10  res.send("Hacked by Middleware 1")
11  // next()
12 })
13
14 // Middleware 2
15 app.use((req, res, next) => {
16   console.log('m2')
17   next()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

node + ×

```
at Function.process_params (C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 90\node_modules\express\lib\router\index.js:346:12)
[nodemon] restarting due to changes...
[nodemon] starting `node main.js`
Example app listening on port 3000
m1
```

Ln 11, Col 14 Spaces: 4 UTF-8 CRLF {} JavaScript

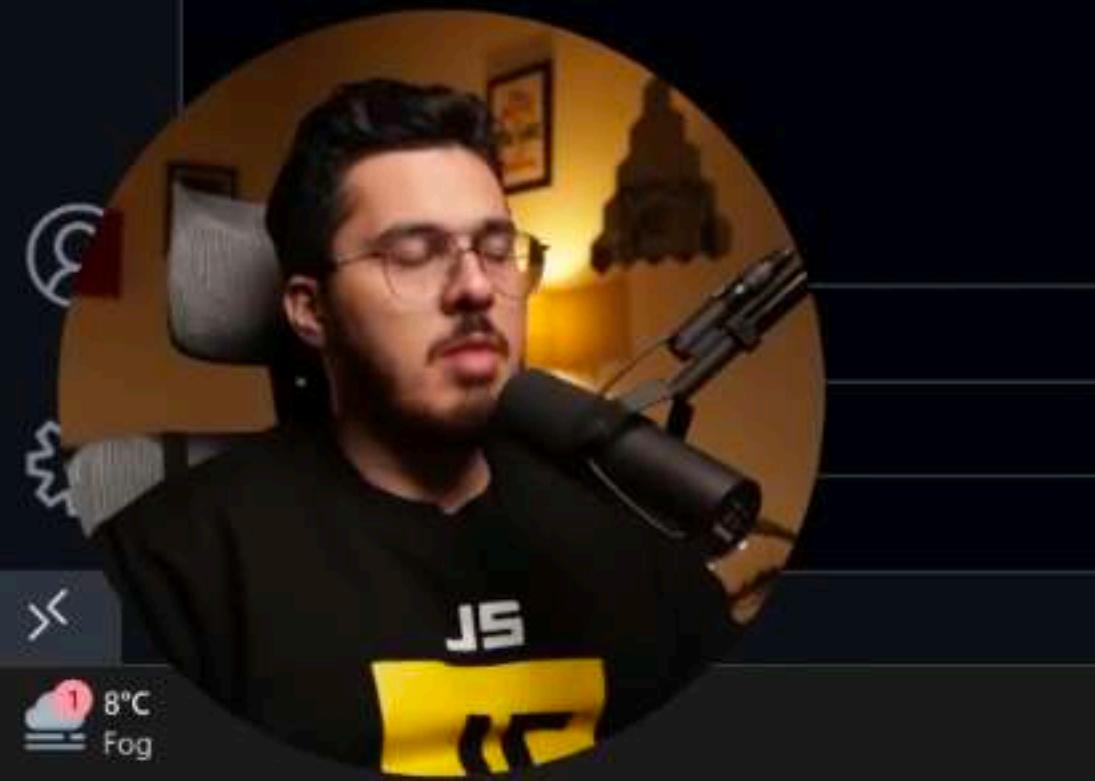
18°C Fog

Search

02:44 Thu

ENG IN

Subscribed





File Edit Selection View Go ... ← → Video 90 □ □ □ □ - □ ×

EXPLORER ... JS main.js X logs.txt node

VIDEO 90

node_modules

public

logs.txt

JS main.js

Middleware.png

package-lock.json

package.json

```
JS main.js > app.use() callback
1 const express = require('express')
2 const app = express()
3 const port = 3000
4 const fs = require("fs")
5
6 // app.use(express.static("public"))
7
8 // Middleware 1 - Logger for our application
9 app.use((req, res, next) => {
10   fs.writeFileSync("logs.txt", `${Date.now()} is a ${req.method}`)
11   console.log(`${Date.now()} is a ${req.method}`)
12   // res.send("Hacked by Middleware 1")
13   next()
14 })
15
16 // Middleware 2
17 app.use((req, res, next) => {
18   console.log('m2')
19   next()
20 })
21
22 app.get('/', (req, res) => {
23   res.send('Hello World!')
24 })
25
26 app.get('/about', (req, res) => {
27   res.send('Hello about!')
28 })
```

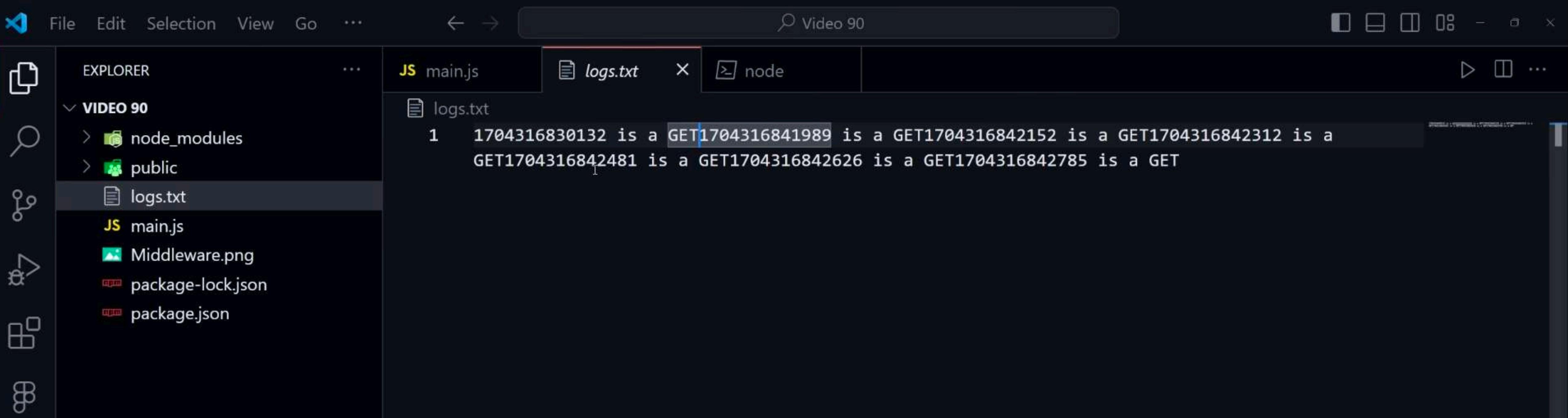
Ln 12, Col 41 Spaces: 4 UTF-8 CRLF {} JavaScript

19°C Fog

Subscribe

Windows Search Calendar Chrome S4 ✓ Settings File Explorer Google Chrome

02:50 Thu



File Edit Selection View Go ... ← → Video 90 □ □ □ 0% - □ ×

EXPLORER ...

VIDEO 90

- > node_modules
- > public
- logs.txt
- main.js
- Middleware.png
- package-lock.json
- package.json

logs.txt

1 1704316830132 is a GET1704316841989 is a GET1704316842152 is a GET1704316842312 is a GET1704316842481 is a GET1704316842626 is a GET1704316842785 is a GET



Ln 1, Col 23 Spaces: 4 CRLF Plain Text

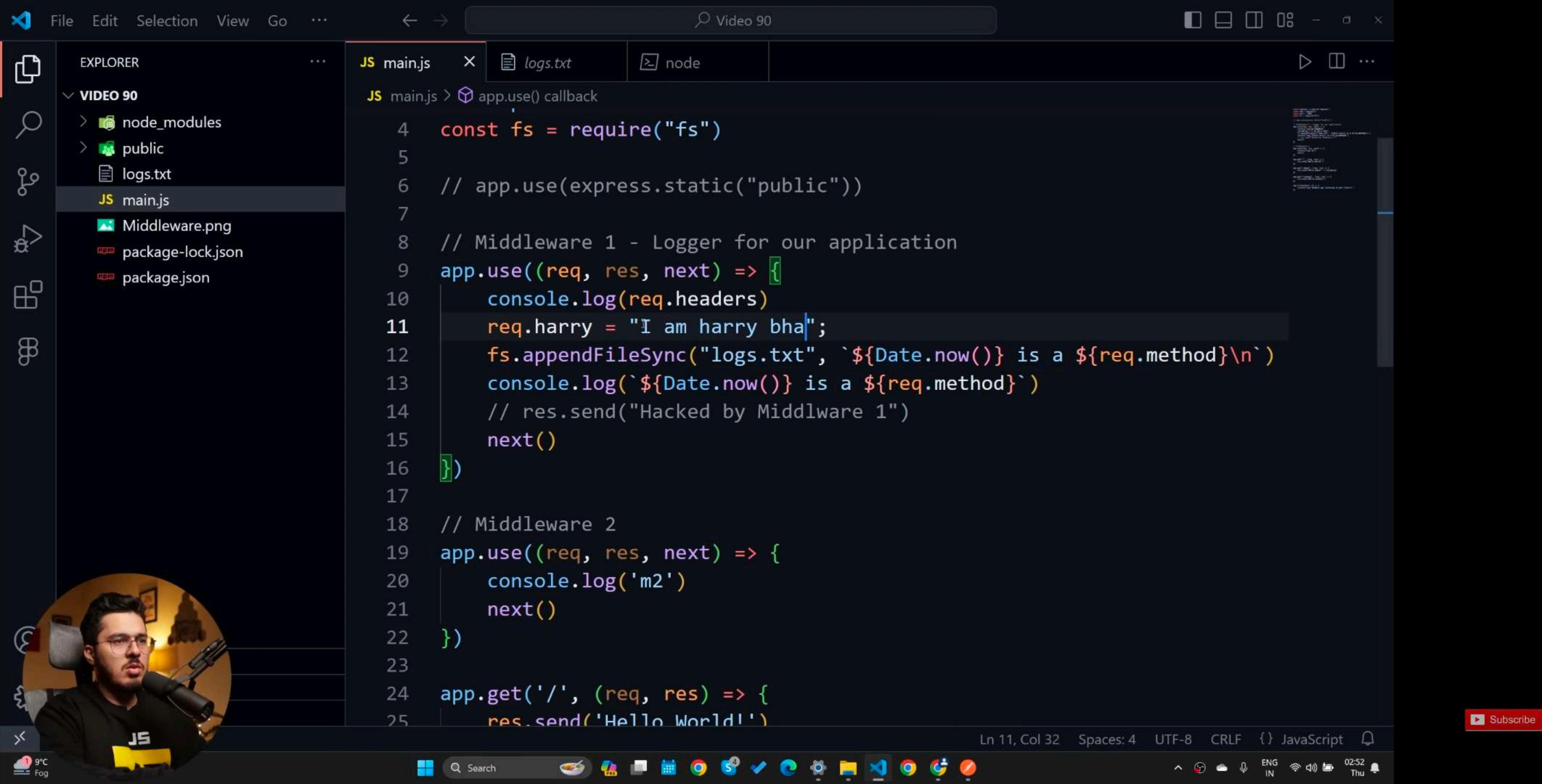
Subscribe

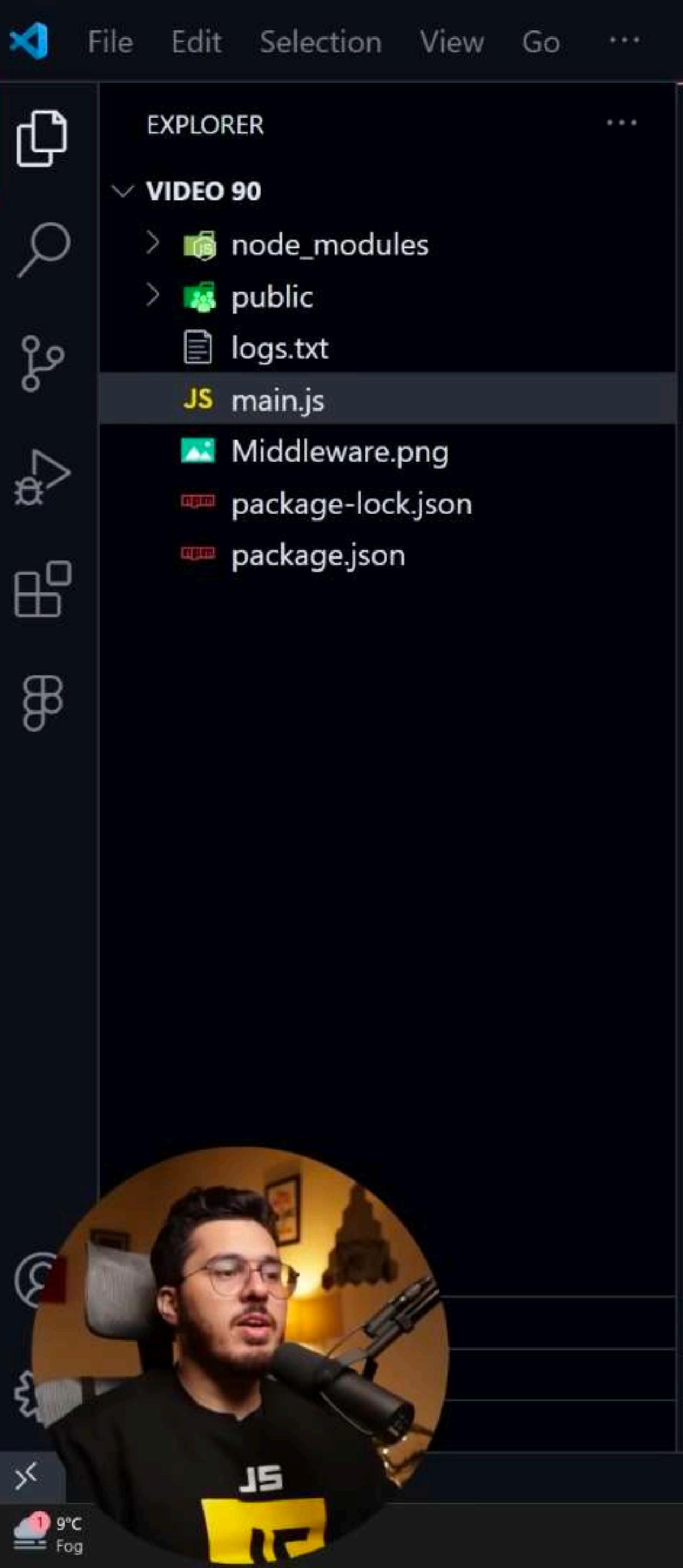


19°C



02:50 Thu





File Edit Selection View Go ... ← → 🔍 Video 90

EXPLORER ...

VIDEO 90

- > node_modules
- > public
- logs.txt
- JS main.js**
- Middleware.png
- package-lock.json
- package.json

JS main.js x logs.txt node

JS main.js > app.use() callback

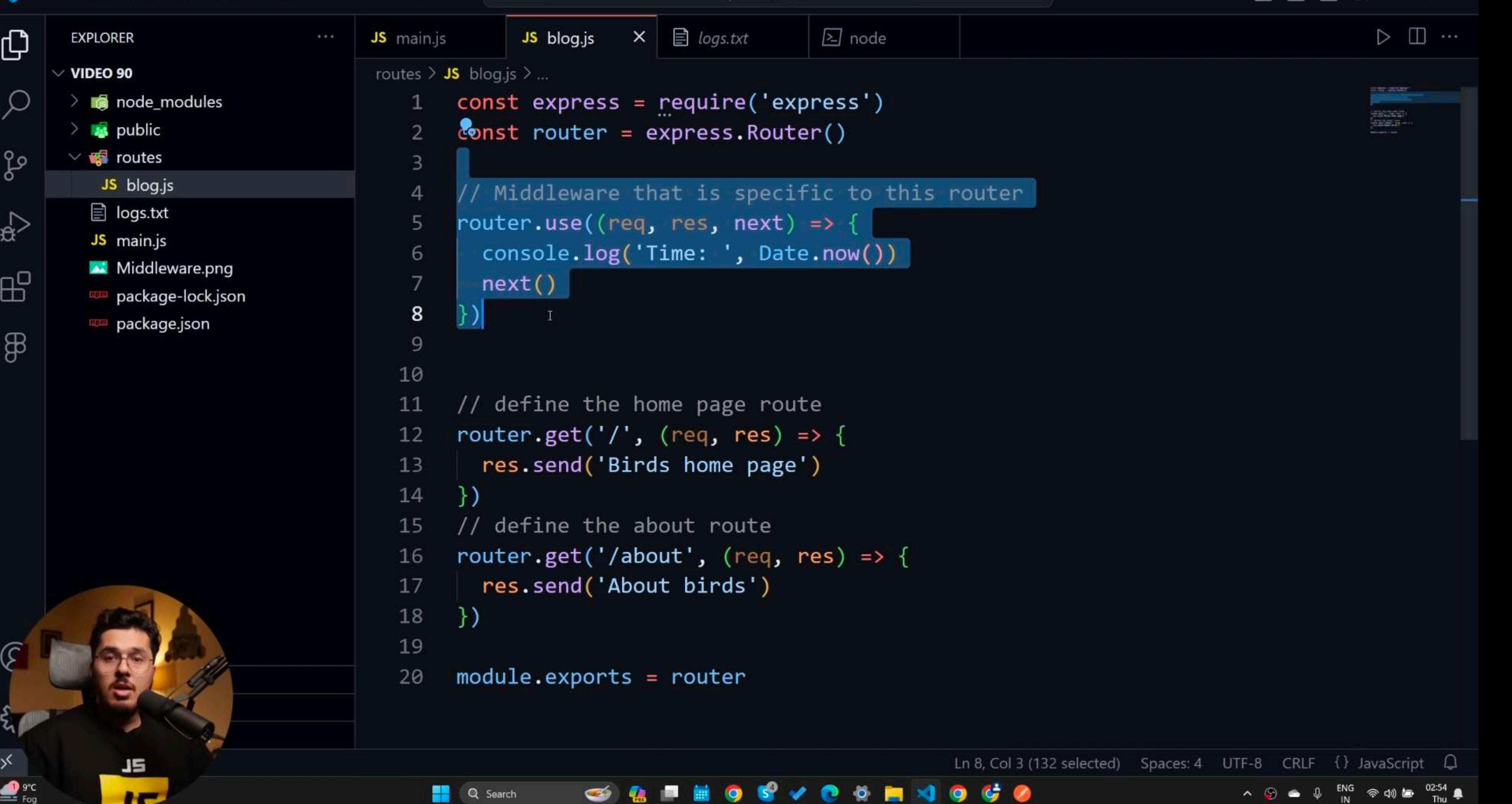
```
17
18 // Middleware 2
19 app.use((req, res, next) => {
20   console.log('m2')
21   next()
22 })
23
24 app.get('/', (req, res) => {
25   res.send('Hello World!')
26 })
27
28 app.get('/about', (req, res) => {
29   res.send('Hello about!' + req.harry)
30 })
31
32 app.get('/contact', (req, res) => {
33   res.send('Hello contact!')
34 })
35
36 app.listen(port, () => {
37   console.log(`Example app listening on port ${port}`)
38 })
```

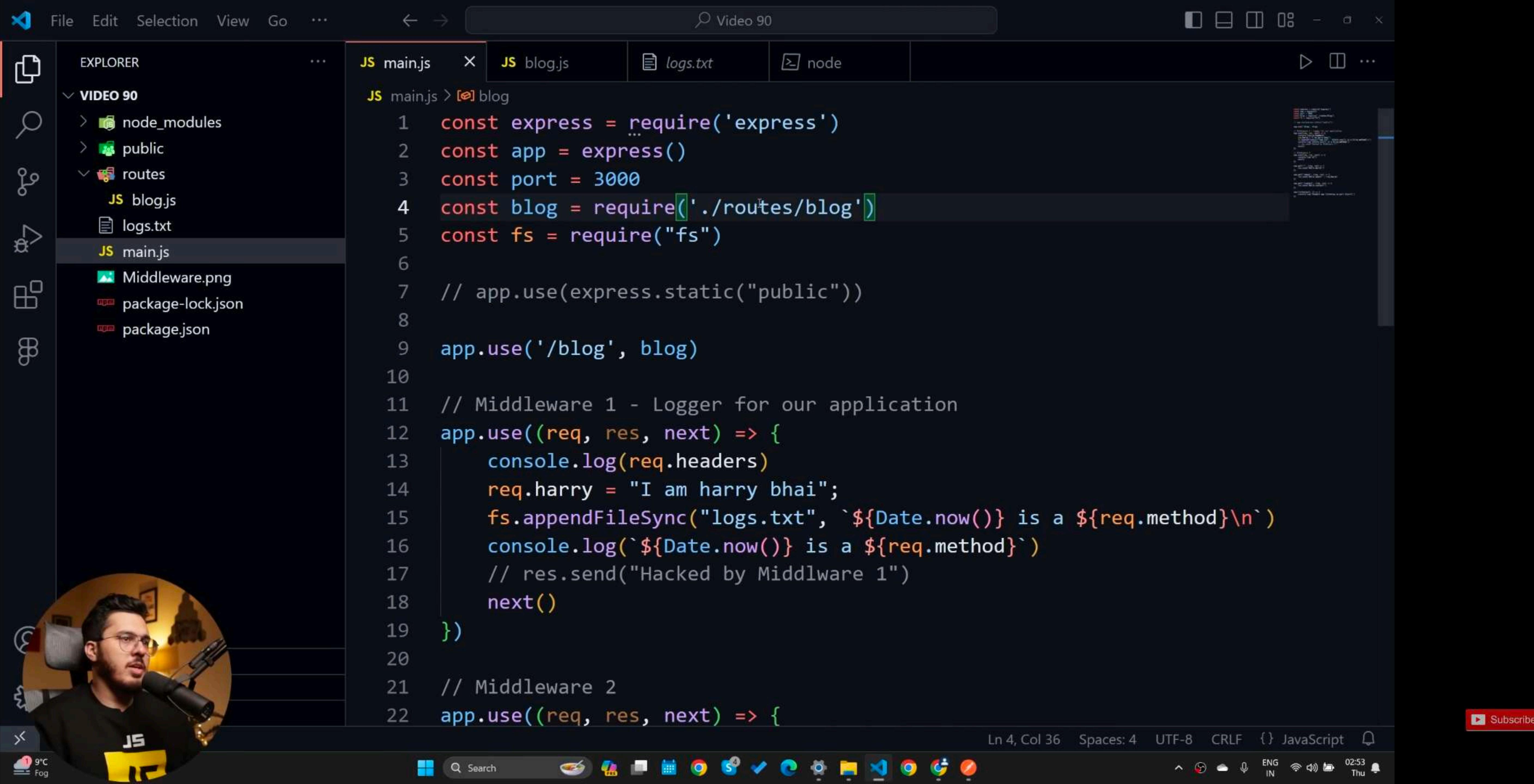
Ln 11, Col 33 Spaces: 4 UTF-8 CRLF {} JavaScript

19°C Fog

Subscribe

Windows Search Calendar Chrome S4 Task View Edge Settings File Explorer Google Chrome





Sigma Web Development Course - W | Writing middleware for use in Express | Using Express middleware | Using Express middleware | localhost:3000/about | +

expressjs.com/en/guide/using-middleware.html

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C...

Black Lives Matter. Support the Equal Justice Initiative.

Express

search Home Getting started Guide API reference Advanced topics Resources

Using middleware

Express is a routing and middleware web framework that has minimal functionality of its own: An Express application is essentially a series of middleware function calls.

Middleware functions are functions that have access to the `request object` (`req`), the `response object` (`res`), and the next middleware function in the application's request-response cycle. The next middleware function is commonly denoted by a variable named `next`.

Middleware functions can perform the following tasks:

- Execute any code.
- Make changes to the request and the response objects.
- End the request-response cycle.
- Call the next middleware function in the stack.

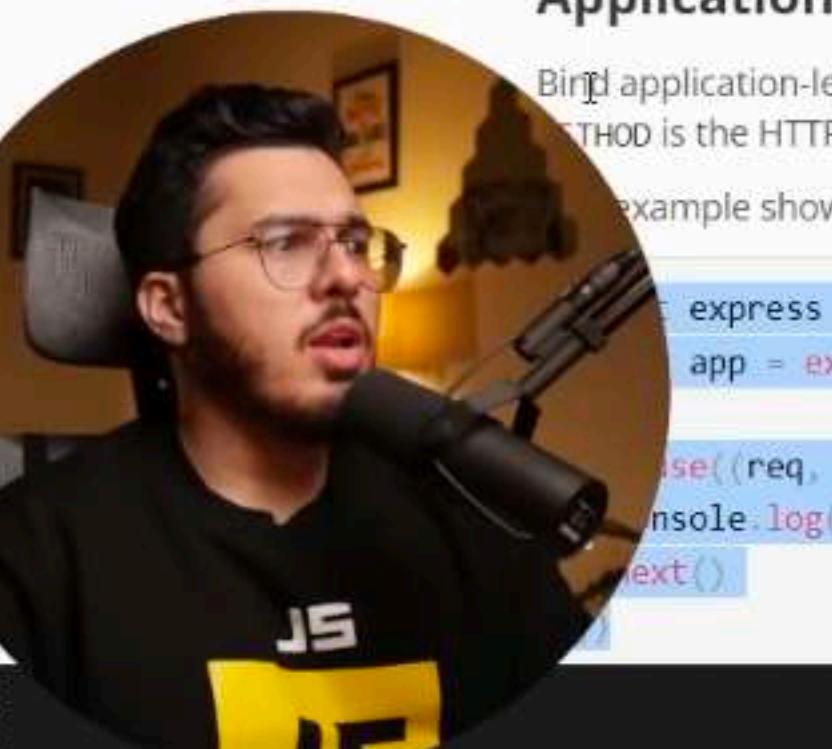
If the current middleware function does not end the request-response cycle, it must call `next()` to pass control to the next middleware function. Otherwise, the request will be left hanging.

An Express application can use the following types of middleware:

- Application-level middleware
- Router-level middleware
- Error-handling middleware
- Built-in middleware
- Third-party middleware

You can load application-level and router-level middleware with an optional mount path. You can also load a series of middleware functions together, which creates a sub-stack of the middleware system at a mount point.

Application-level middleware



```
const express = require('express')
app = express()

use((req, res, next) => [
  console.log('Time:', Date.now())
  next()
])
```

Bind application-level middleware to an instance of the `app object` by using the `app.use()` and `app.METHOD()` functions, where `METHOD` is the HTTP method of the request that the middleware function handles (such as `GET`, `PUT`, or `POST`) in lowercase.

The following example shows a middleware function with no mount path. The function is executed every time the app receives a request.

19°C Fog

Search

02:56 Thu

ENG IN

Subscribe

7:48

38 KB/s 5G 37%



Join CodeWithHarry's Da...

Sponsored · codewithharry.c...

Buy now

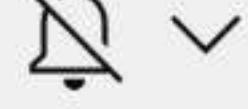
ejs Template Engine in Express | Sigma Web Development Course - Tutorial #92

108K views 1y ago Sigma Web Development Course - Web ...more



CodeWithHarry 8.37M

Join



3.9K



Share

Remix

Download

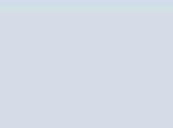
Comments 579



Please guys support him and write Sigma batch op

 HOSTINGER.IN

Change starts with .org



End of playlist

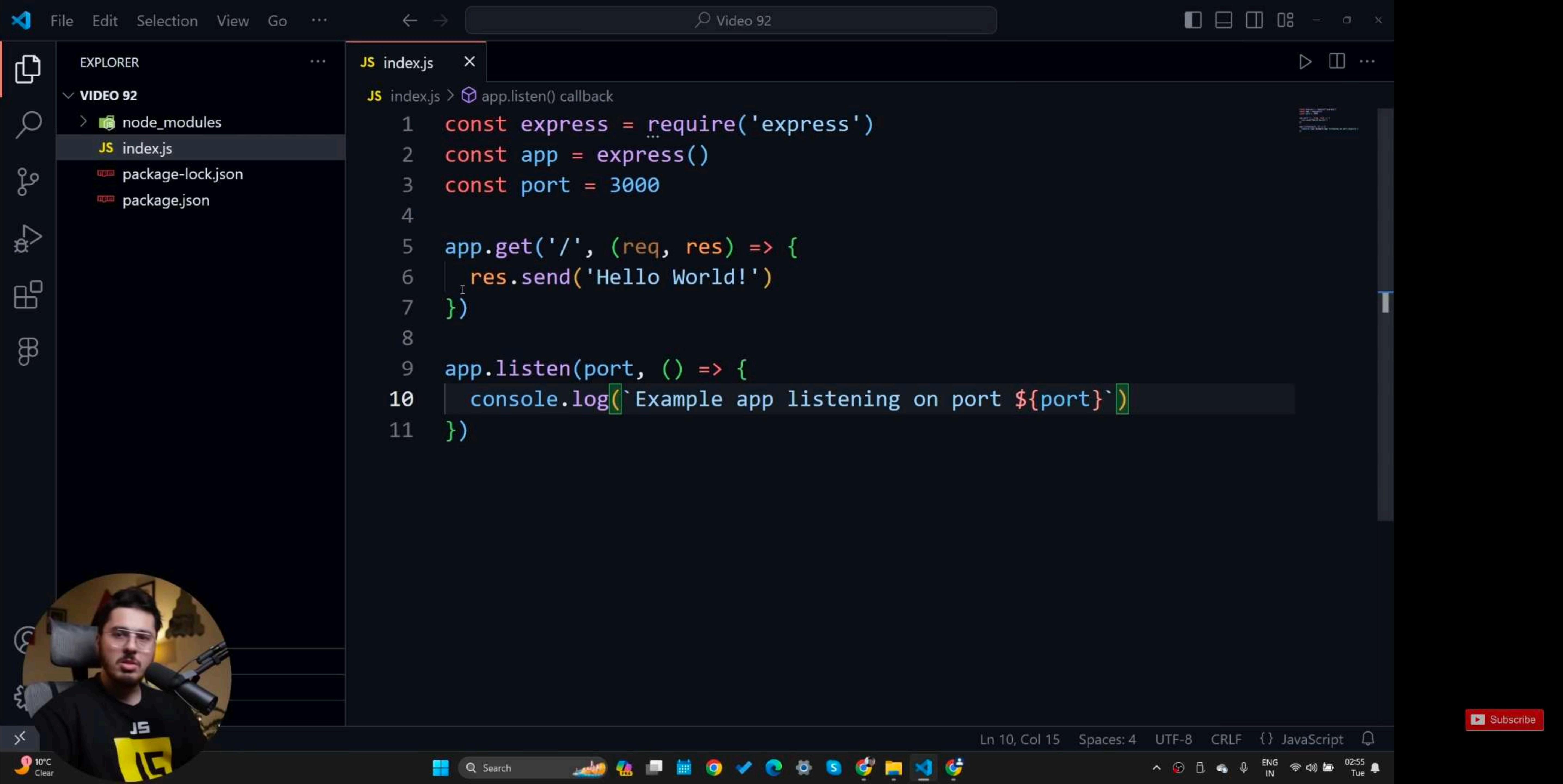


Today's Special • 4/4



Claim your .org domain from
Hostinger





Sigma Web Development Course - W... X | Express "Hello World" example X | localhost:3000 X | Bootstrap - The most popular HTML, CSS, and JS...

getbootstrap.com

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C...

Docs Examples Icons Themes Blog

Search CTRL K

New in v5.3 Color mode support, expanded color palette, and more!

B

Build fast, responsive sites with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

\$ npm i bootstrap@5.3.2

Read the docs

Currently v5.3.2 · [Download](#) · [All releases](#)

Get started any way you want

Sponsored • 1 of 2

Waiting for www...

10°C Clear

Subscribe

ENG IN 02:55 Tue

Sigma Web Development Course - W | Express "Hello World" example | localhost:3000 | B Get started with Bootstrap - Bootstrap | +

getbootstrap.com/docs/5.3/getting-started/introduction/

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C...

Getting started

Introduction

- Download
- Contents
- Browsers & devices
- JavaScript
- Webpack
- Parcel
- Vite
- Accessibility
- RFS
- RTL
- Contribute

Customize

- Overview
- Sass
- Options
- Color
- Color modes
- Components
- CSS variables
- Optimize



Layout

- Breakpoints
- Containers
- Grid
- Columns
- Gutters
- Utilities

CDN links

As reference, here are our primary CDN links.

```
</body>
</html>
```

2. **Include Bootstrap's CSS and JS.** Place the `<link>` tag in the `<head>` for our CSS, and the `<script>` tag for our JavaScript bundle (including Popper for positioning dropdowns, poppers, and tooltips) before the closing `</body>`. Learn more about our [CDN links](#).

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
  </body>
</html>
```

You can also include [Popper](#) and our JS separately. If you don't plan to use dropdowns, popovers, or tooltips, save some kilobytes by not including Popper.

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-5RqzCQVHP9qKmekw1Vzq+u7NfWjZJw5WFXdkZK5" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.min.js" integrity="sha384-bfQe4YF50K6KkZq+u7NfWjZJw5WFXdkZK5" crossorigin="anonymous"></script>
```

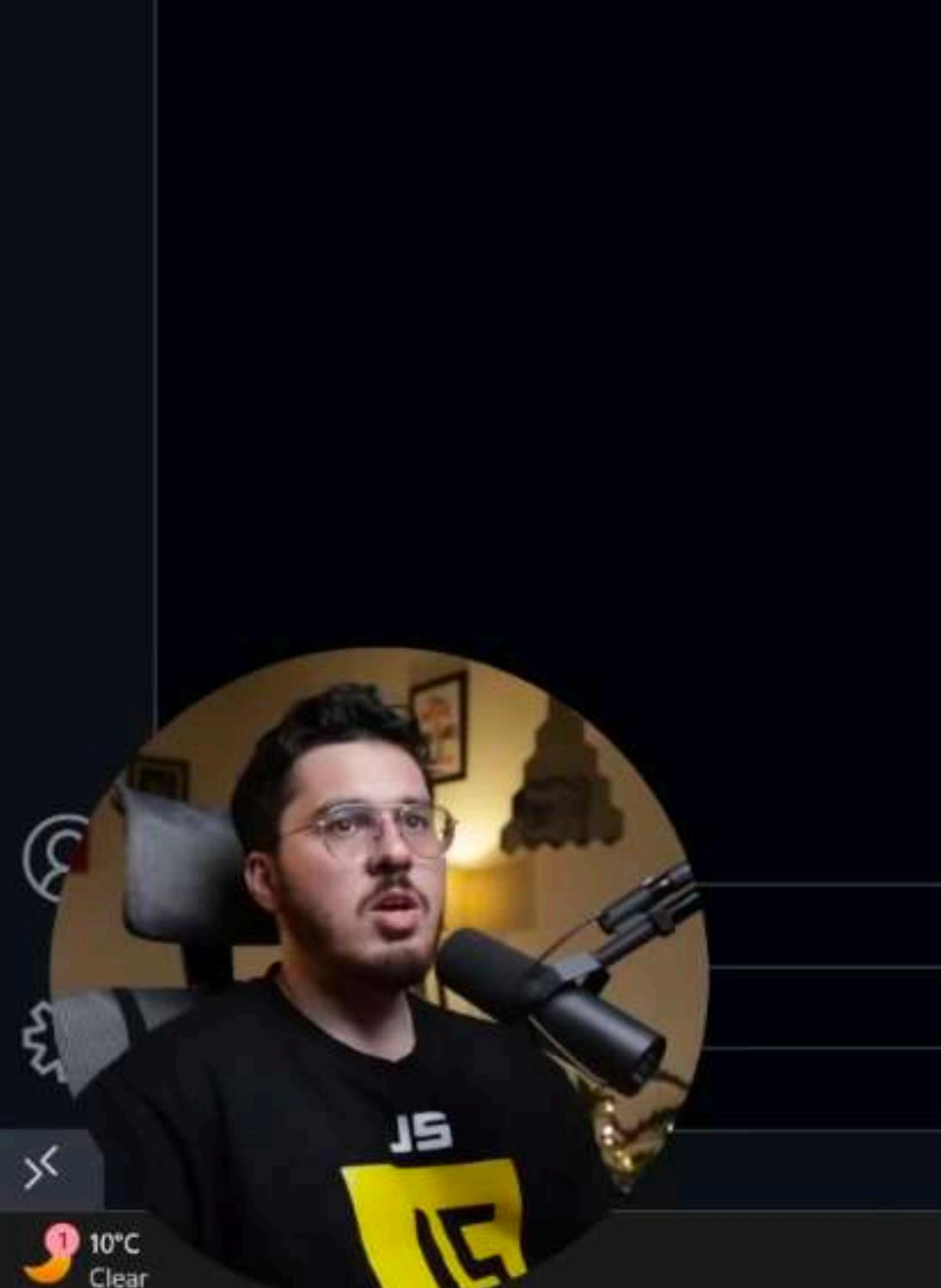
3. **Hello, world!** Open the page in your browser of choice to see your Bootstrapped page. Now you can start building with Bootstrap by creating your own [layout](#), adding dozens of [components](#), and utilizing [our official examples](#).

On this page

- Quick start
- CDN links
- Next steps
- JS components
- Important globals
- HTML5 doctype
- Viewport meta
- Box-sizing
- Reboot
- Community

Copied!

Subscribe



File Edit Selection View Go ... ← → 🔍 Video 92

EXPLORER ... JS index.js index.html X

VIDEO 92

node_modules

templates

index.html

index.js

package-lock.json

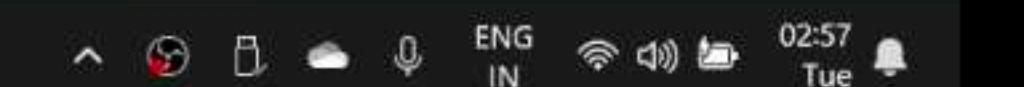
package.json

templates > index.html > html

```
1  <!doctype html>
2  <html lang="en">
3  <head>
4  <meta charset="utf-8">
5  <meta name="viewport" content="width=device-width, initial-scale=1">
6  <title>Bootstrap demo</title>
7  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet"
8  integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzcDSCmG1MXxSR1GAsXEV/Dwykc2MPK8M2HN" crossorigin="anonymous">
9  </head>
10 <body>
11 <h1>Hello, world!</h1>
12 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
13  integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>
</body>
</html>
```

Ln 13, Col 8 Spaces: 4 UTF-8 CRLF HTML

Subscribe



getbootstrap.com/docs/5.3/components/navbar/

Docs Examples Icons Themes Blog Search CTRL K

On this page

How it works Supported content

Brand

Text

Image

Image and text

Nav

Forms

Text

Color schemes

Containers

Placement

Scrolling

Responsive behaviors

Toggler

External content

Offcanvas

CSS

Variables

Sass variables

Sass loops

Navbar Home Link Dropdown ▾ Disabled Search Search Copied!

HTML

```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navSupportedContent">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown">
            Dropdown
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Action</a></li>
            <li><a class="dropdown-item" href="#">Another action</a></li>
            <li><hr class="dropdown-divider"></li>
            <li><a class="dropdown-item" href="#">Something else here</a></li>
          </ul>
        </li>
        <li class="nav-item">
```

10°C Clear

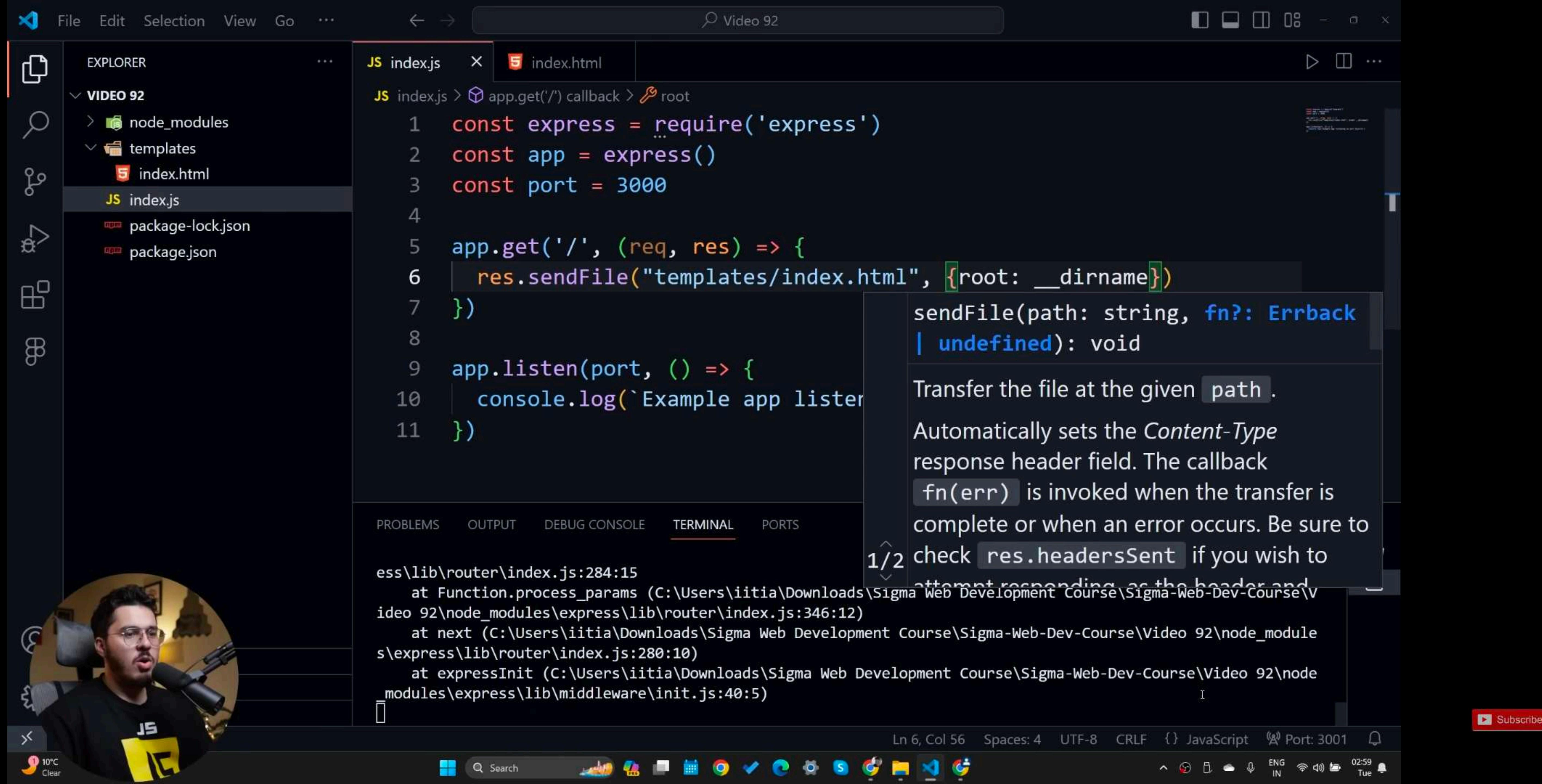
Subscribe



10°C Clear

Search

0 ENG IN 02:57 Tue



Sigma Web Development Co | Express "Hello World" example | Bootstrap demo | B Navbar · Bootstrap v5.3 | B Accordion · Bootstrap v5.3 | B Alerts · Bootstrap v5.3 | B Badges · Bootstrap v5.3 | B Breadcrumb · Bootstrap v5.3 | EJS -- Embedded JavaScript

ejs.co

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C...

EJS

About Features Get Started Docs Support License GitHub

Jake
The original JavaScript build tool.

<%= EJS %>

Embedded JavaScript templating.

[GET STARTED](#)

What is EJS?



What is the "E" for? "Embedded?" Could be. How about "Effective," "Elegant," or just "Easy"? EJS is a simple templating language that lets you generate HTML markup with plain JavaScript. No religiousness about how to organize things. No reinvention of iteration and control-flow. It's just plain JavaScript.

 **Use plain JavaScript**
We love JavaScript. It's a totally friendly language. All templating languages grow to be Turing-complete. Just cut out the

 **Fast development time**
Don't waste time and attention figuring out arcane new syntax because 'elegance' — or how to preprocess your data so it will

 **Simple syntax**
JavaScript code in simple, straightforward scriptlet tags. Just write JavaScript that emits the HTML you want, and get the job

10°C Clear

Search

0 ENG IN 03:02 Tue

Subscribe

EJS

About Features **Get Started** Docs Support License GitHub

Get Started

Install

It's easy to install EJS with NPM.

```
$ npm install ejs
```

Use

Pass EJS a template string and some data. BOOM, you've got some HTML.

```
let ejs = require('ejs');
let people = ['geddy', 'neil', 'alex'];
let html = ejs.render('<%= people.join(", "); %>', {people: people});
```

CLI

Feed it a template file and a data file, and specify an output file.

```
ejs ./template_file.ejs -f data_file.json -o ./output.html
```

Browser support

Download a browser build from the latest release, and use it in a script tag.

```
<script src="ejs.js"></script>
<script>
  let people = ['geddy', 'neil', 'alex'];
  let html = ejs.render('<%= people.join(", "); %>', {people: people});
</script>
```



10°C Clear

Search

Subscribed

03:02 Tue

ENG IN



File Edit Selection View Go ... ← → Video 92

EXPLORER

VIDEO 92

- node_modules
- templates
- index.html
- index.js
- package-lock.json
- package.json

JS index.js x index.html

JS index.js > app.get('/blog/:slug') callback

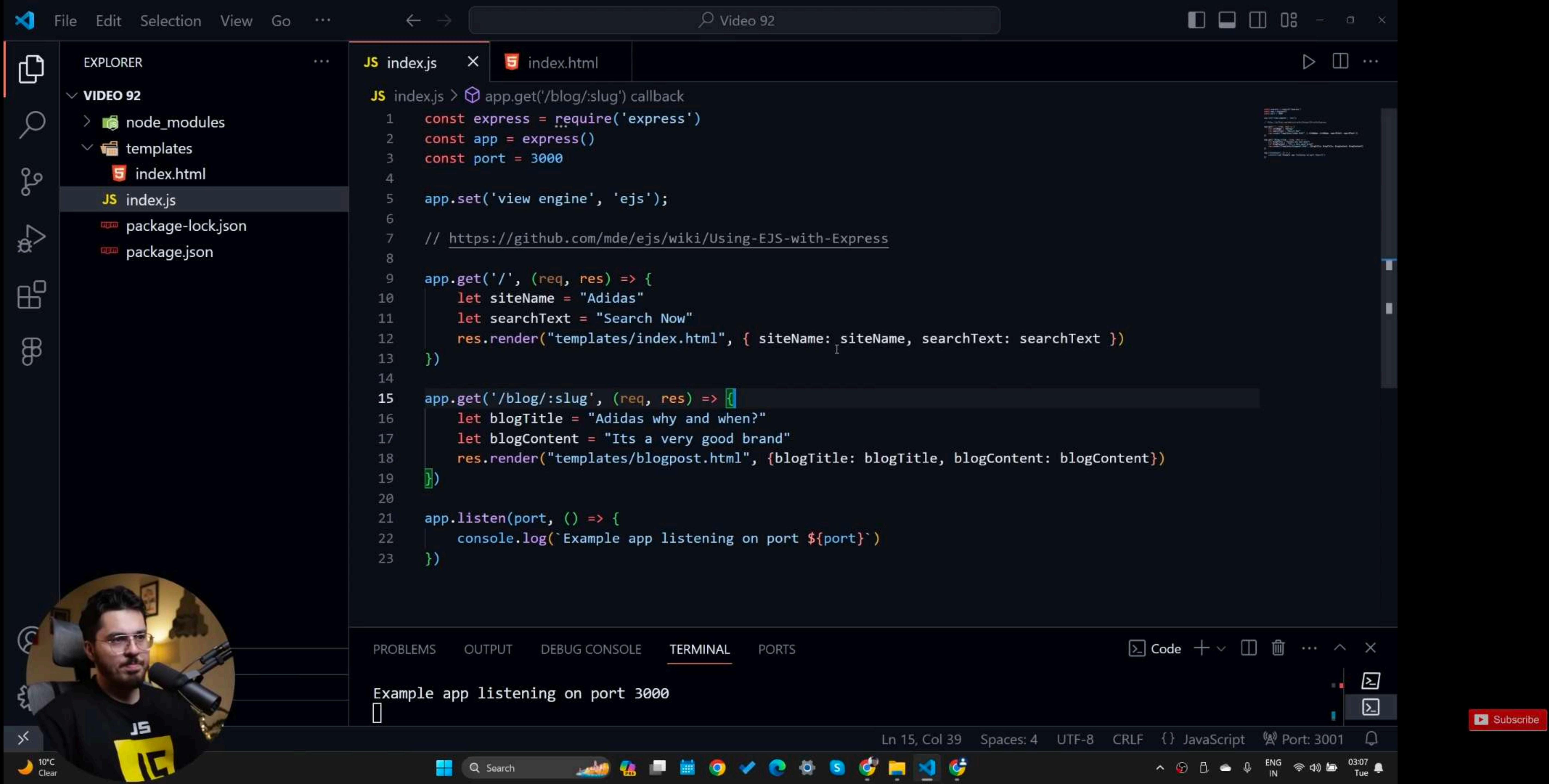
```
1 const express = require('express')
2 const app = express()
3 const port = 3000
4
5 app.get('/', (req, res) => {
6   let siteName = "Adidas"
7   let searchText = "Search Now"
8   res.sendFile("templates/index.html", { root: __dirname })
9 }
10
11 app.get('/blog/:slug', (req, res) => {
12   let blogTitle = "Adidas why and when?"
13   let blogContent = "The is a very good brand"
```

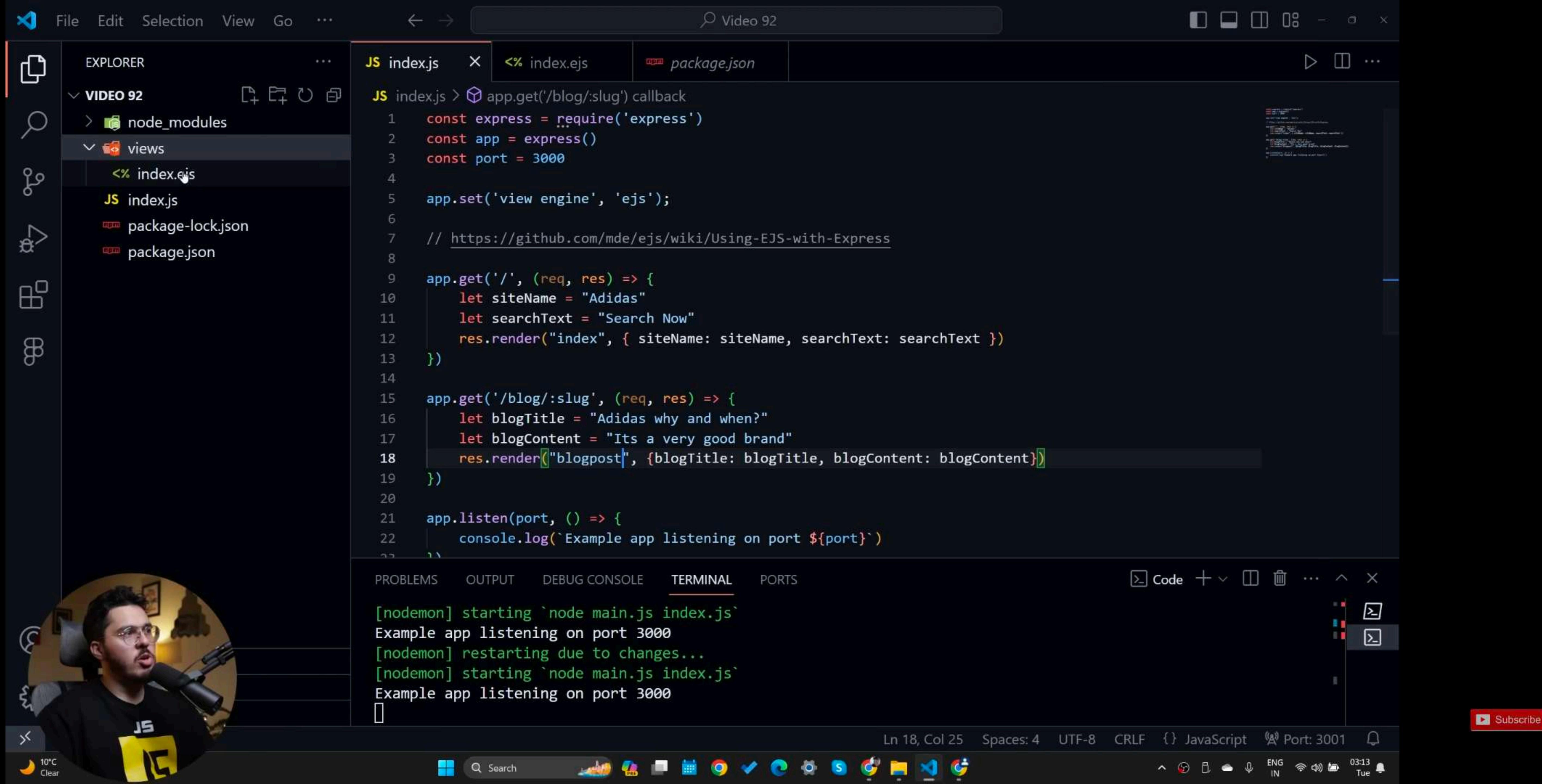
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

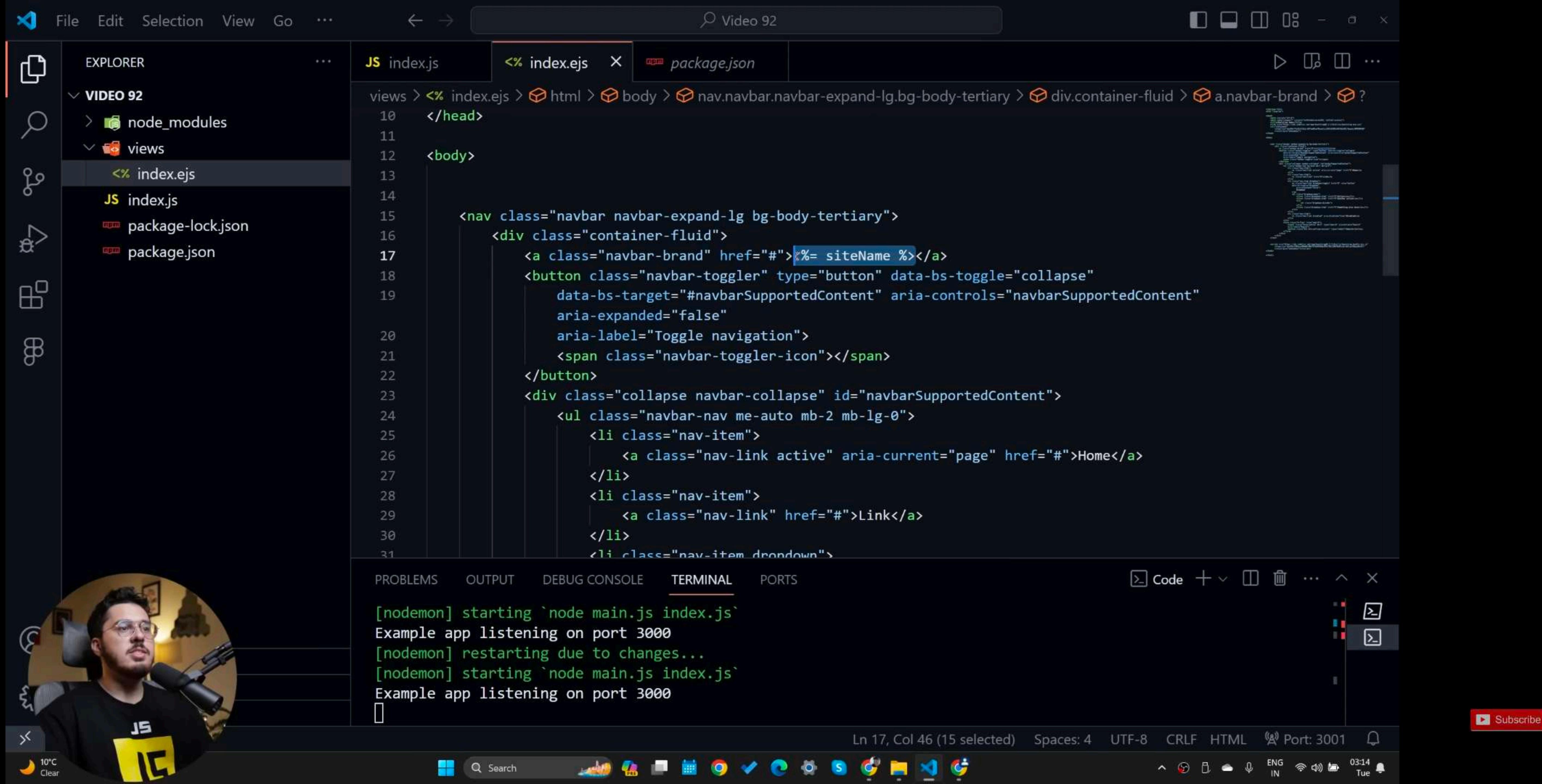
Example app listening on port 3000

- PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 92> ^C
- PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 92> ^C
- PS C:\Users\iitia\Downloads\Sigma Web Development Course\Sigma-Web-Dev-Course\Video 92> npm install ejs

[.....] - idealTree:Video 92: sill idealTree buildDeps







Sigma Web Dev | Express Hello | Bootstrap dem... | Error: Cannot f... | node.js - How | javascript - No | Error: Cannot f... | B Navbar - Boot... | B Accordion - Bo... | B Alerts - Boot... | B Badges - Boot... | B Breadcrumb - | EJS - Embedd... | +

ejs.co/#install

Sigma Web Developme... GitHub - CodeWithHarr... Learn to code online - C...

EJS

About Features Get Started Docs **Support** License GitHub

- `-d / --debug` Outputs generated function body
- `-h / --help` Display this help message.
- `-V/v / --version` Display the EJS version.

Some examples of use:

```
$ ejs -p [ -c ] ./template_file.ejs -o ./output.html
$ ejs ./test/fixtures/user.ejs name=Lerxst
$ ejs -n -l _ ./some_template.ejs -f ./data_file.json
```

Custom delimiters

Custom delimiters can be applied on a per-template basis, or globally:

```
let ejs = require('ejs'),
  users = ['geddy', 'neil', 'alex'];

// Just one template
ejs.render('<?= users.join(" | "); ?>', {users: users},
  {delimiter: '?'});
// => 'geddy | neil | alex'

// Or globally
ejs.delimiter = '$';
ejs.render('<$= users.join(" | "); $>', {users: users});
// => 'geddy | neil | alex'
```

Caching

EJS ships with a basic in-process cache for caching the intermediate JavaScript functions used to render templates. It's easy to plug in LRU caching using Node's `lru-cache` library:

```
let ejs = require('ejs'),
  LRU = require('lru-cache');
ejs.cache = LRU(100); // LRU cache with 100-item limit
```

If you want to clear the EJS cache, call `ejs.clearCache`. If you're using the LRU cache and need a different limit, `lruCache.maxSize` is probably a better name for the LRU limit.



10°C Clear

Subscribe

