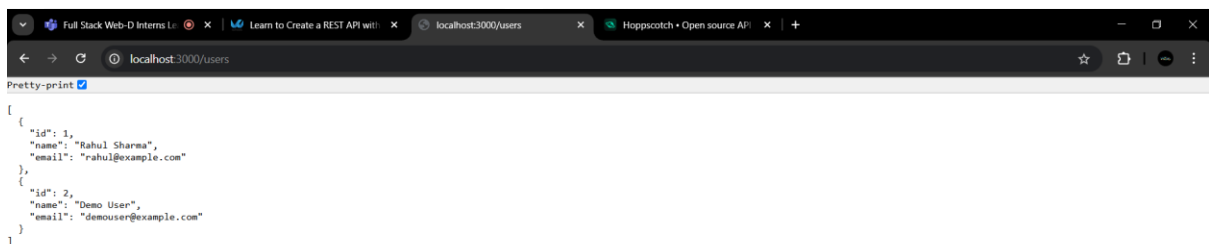
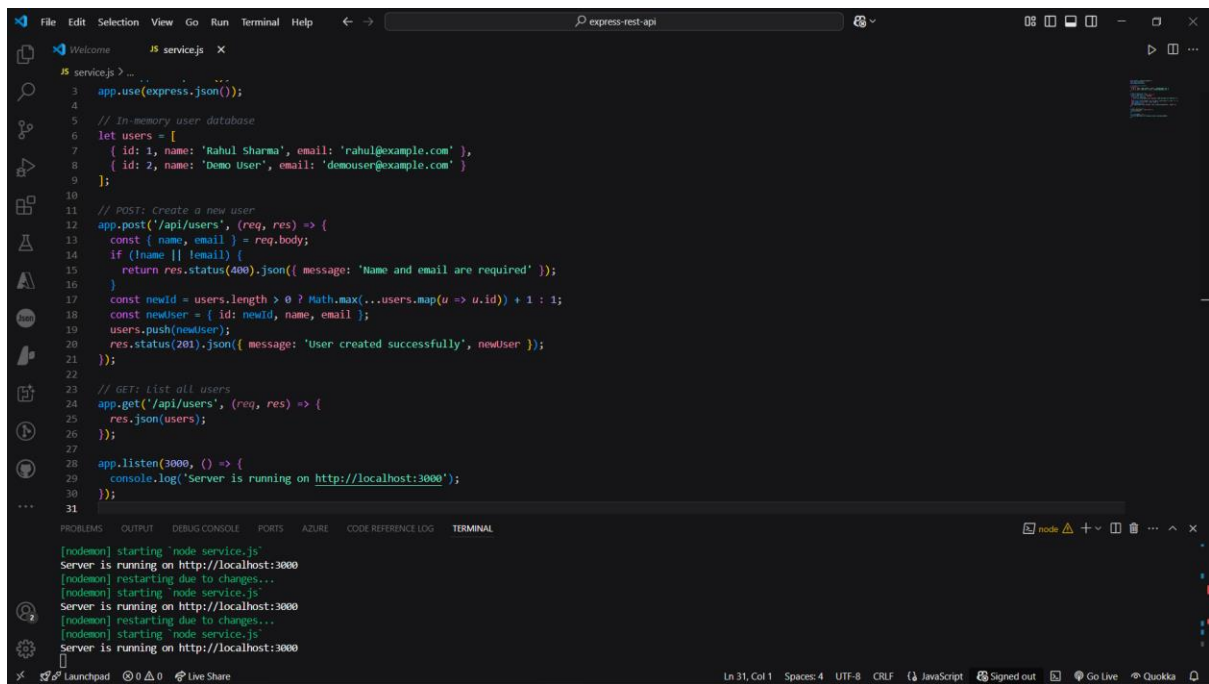
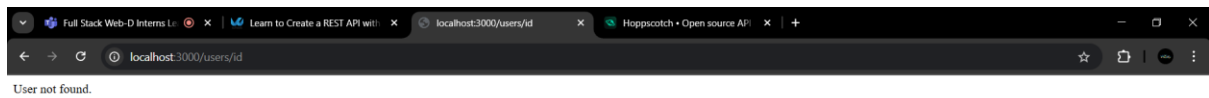


11 June 2025 – DAY 9

SESSION 1(11AM – 2PM)

Topic:- Express.js REST API





Full Stack Web-D Interns L... x Learn to Create a REST API with... x localhost:3000/users x Hoppscotch • Open source API... x +

hoppscotch.io

HOPPSCOTCH

Try "Create workspace" Ctrl K

Save My Workspace Login

POST Untitled +

Select environment

POST http://localhost:3000/api/users Send Save

Parameters Body Headers Authorization Pre-request Script Tests Variables

Content Type application/json Override

Raw Request Body

```
1 {
2   "name": "vaishnavi",
3   "email": "abc@example.com"
4 }
```

Status: 201 • Created Time: 10 ms Size: 339 B

JSON Raw Headers Test Results

Response Body

```
1 {
2   "message": "User created successfully",
3   "newUser": {
4     "id": 3,
5     "name": "vaishnavi",
6     "email": "abc@example.com"
7   }
8 }
```

Personal Workspace > Collections

Search

+ New

Collections are empty

Import or create a collection

Import

+ Add new

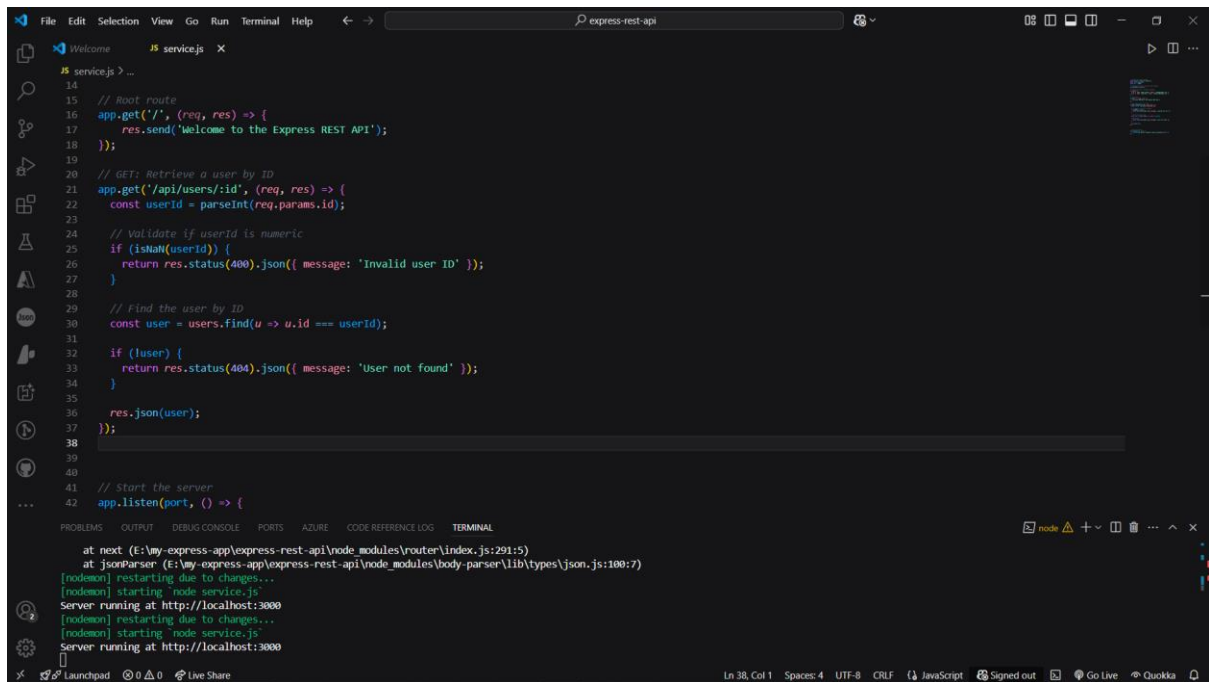
Help & feedback

Full Stack Web-D Interns L... x Learn to Create a REST API with... x localhost:3000/users x Hoppscotch • Open source API... x +

localhost:3000/users

Pretty-print

```
[
  {
    "id": 1,
    "name": "Rahul Sharma",
    "email": "rahul@example.com"
  },
  {
    "id": 2,
    "name": "Demo User",
    "email": "demouser@example.com"
  },
  {
    "id": 3,
    "name": "vaishnavi",
    "email": "abc@example.com"
  }
]
```

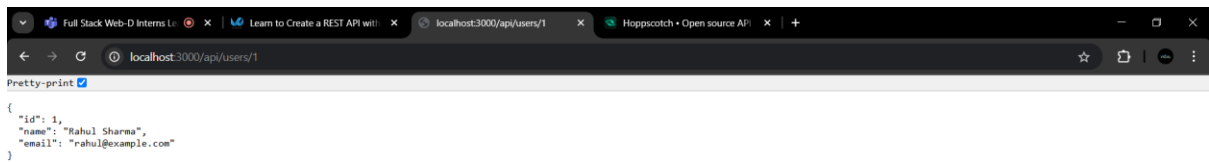


The image shows a Visual Studio Code editor window with a file named `service.js` open. The code is a REST API service using Express.js. It includes a root route that returns a welcome message, a GET route for retrieving a user by ID, and a server listening on port 3000. The terminal output shows the server starting successfully and running at `http://localhost:3000`.

```
14 // Root route
15 // Root route
16 app.get('/', (req, res) => {
17   res.send('Welcome to the Express REST API');
18 });
19
20 // GET: Retrieve a user by ID
21 app.get('/api/users/:id', (req, res) => {
22   const userId = parseInt(req.params.id);
23
24   // Validate if userId is numeric
25   if (isNaN(userId)) {
26     return res.status(400).json({ message: 'Invalid user ID' });
27   }
28
29   // Find the user by ID
30   const user = users.find(u => u.id === userId);
31
32   if (!user) {
33     return res.status(404).json({ message: 'User not found' });
34   }
35
36   res.json(user);
37 });
38
39
40
41 // Start the server
42 app.listen(port, () => {
```

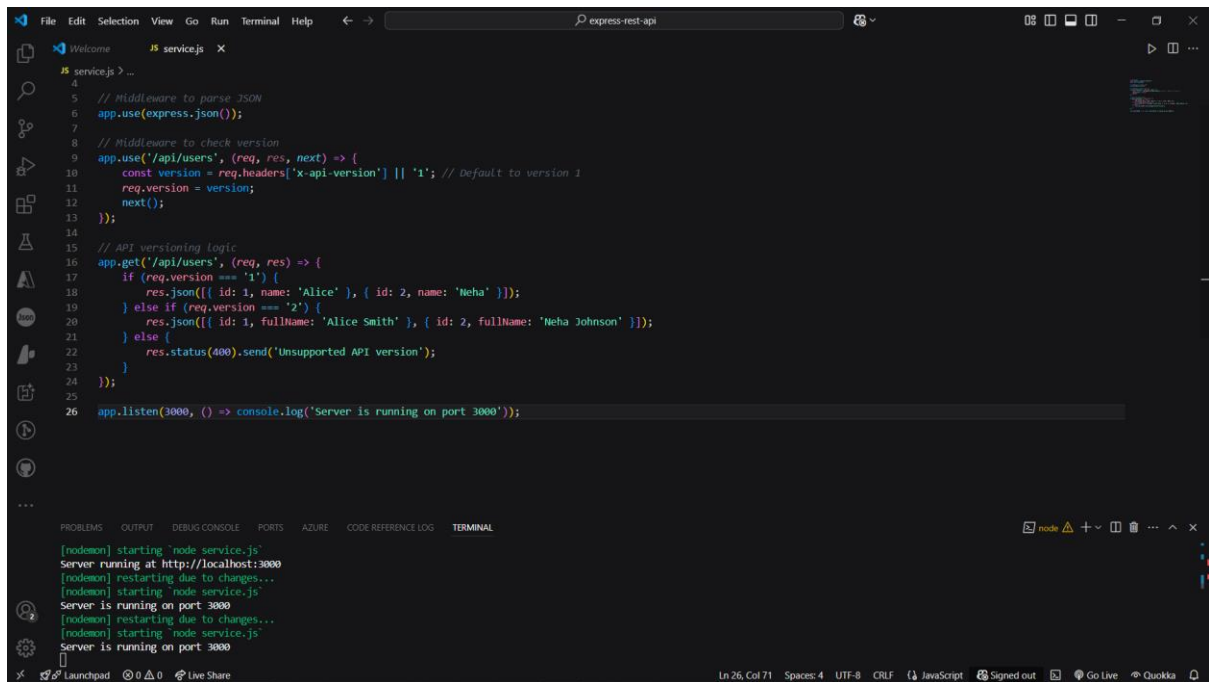
Terminal Output:

```
at next (E:\my-express-app\express-rest-api\node_modules\router\index.js:291:5)
at jsonParser (E:\my-express-app\express-rest-api\node_modules\body-parser\lib\types\json.js:100:7)
[nodemon] restarting due to changes...
[nodemon] starting 'node service.js'
Server running at http://localhost:3000
[nodemon] restarting due to changes...
[nodemon] starting 'node service.js'
Server running at http://localhost:3000
```



The image shows a web browser window with the address bar displaying `localhost:3000/api/users/1`. The page content shows the JSON response from the API, which is a user object with the following properties:

```
{
  "id": 1,
  "name": "Rahul Sharma",
  "email": "rahul@example.com"
}
```

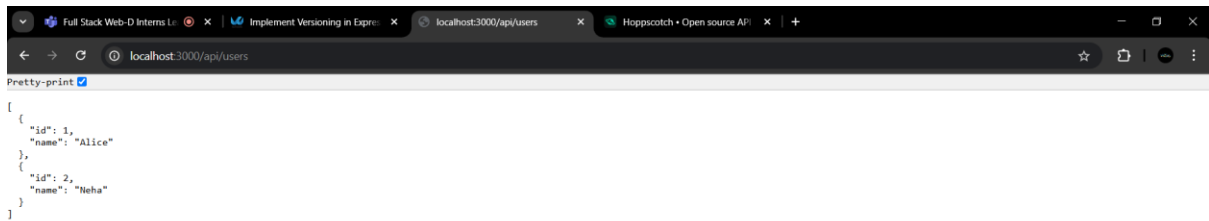


The image shows a Visual Studio Code editor window with a file named `service.js` open. The code implements an Express.js API with versioning. It uses `express.json()` for parsing, and a custom middleware to check the `x-api-version` header. The API endpoint `/api/users` returns different JSON responses based on the version: version 1 returns user details, and version 2 returns full names. The server is configured to listen on port 3000.

```
1 // service.js
2
3 // Middleware to parse JSON
4 app.use(express.json());
5
6 // Middleware to check version
7 app.use('/api/users', (req, res, next) => {
8   const version = req.headers['x-api-version'] || '1'; // Default to version 1
9   req.version = version;
10   next();
11 });
12
13 // API versioning logic
14 app.get('/api/users', (req, res) => {
15   if (req.version === '1') {
16     res.json([
17       { id: 1, name: 'Alice' },
18       { id: 2, name: 'Neha' }
19     ]);
20   } else if (req.version === '2') {
21     res.json([
22       { id: 1, fullName: 'Alice Smith' },
23       { id: 2, fullName: 'Neha Johnson' }
24     ]);
25   } else {
26     res.status(400).send('Unsupported API version');
27   }
28 });
29
30 app.listen(3000, () => console.log('Server is running on port 3000'));
```

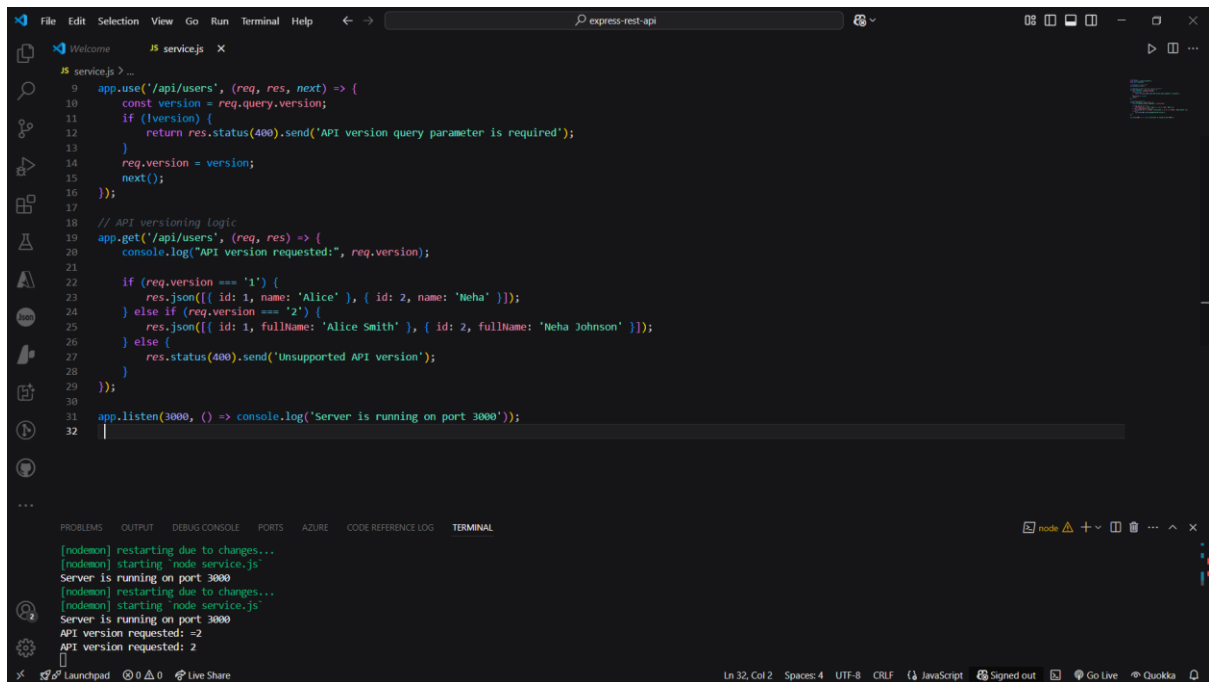
The terminal at the bottom shows the output of running `node service.js`. It indicates that the server is running at `http://localhost:3000` and that it restarted due to changes in the code.

```
[nodemon] starting 'node service.js'
Server running at http://localhost:3000
[nodemon] restarting due to changes...
[nodemon] starting 'node service.js'
Server is running on port 3000
[nodemon] restarting due to changes...
[nodemon] starting 'node service.js'
Server is running on port 3000
```



The image shows a web browser window with the address bar set to `localhost:3000/api/users`. The page content displays the JSON response from the API, which is a pretty-printed array of two user objects.

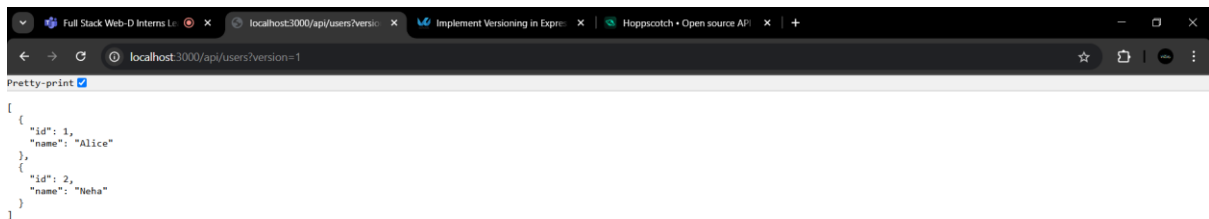
```
[{"id": 1, "name": "Alice"}, {"id": 2, "name": "Neha"}]
```



```
File Edit Selection View Go Run Terminal Help express-rest-api
Welcome JS service.js x
9 app.use('/api/users', (req, res, next) => {
10   const version = req.query.version;
11   if (!version) {
12     return res.status(400).send('API version query parameter is required');
13   }
14   req.version = version;
15   next();
16 });
17
18 // API versioning logic
19 app.get('/api/users', (req, res) => {
20   console.log("API version requested:", req.version);
21
22   if (req.version === '1') {
23     res.json([ { id: 1, name: 'Alice' }, { id: 2, name: 'Neha' } ]);
24   } else if (req.version === '2') {
25     res.json([ { id: 1, fullName: 'Alice Smith' }, { id: 2, fullName: 'Neha Johnson' } ]);
26   } else {
27     res.status(400).send('Unsupported API version');
28   }
29 });
30
31 app.listen(3000, () => console.log('Server is running on port 3000'));
32
```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE CODE REFERENCE LOG TERMINAL

```
[nodemon] restarting due to changes...
[nodemon] starting 'node service.js'
Server is running on port 3000
[nodemon] restarting due to changes...
[nodemon] starting 'node service.js'
Server is running on port 3000
API version requested: -2
API version requested: 2
```



Full Stack Web-D Interns L... localhost:3000/api/users/versi... Implement Versioning in Expre... Hoppscotch • Open source API

localhost:3000/api/users/version=1

Pretty-print

```
{
  "id": 1,
  "name": "Alice"
},
{
  "id": 2,
  "name": "Neha"
}
```

```
Full Stack Web-D Interns L... localhost:3000/api/users?version=2 Implement Versioning in Expre... Hoppscotch • Open source API...  
localhost:3000/api/users?version=2  
Pretty-print  
[  
  {  
    "id": 1,  
    "fullName": "Alice Smith"  
  },  
  {  
    "id": 2,  
    "fullName": "Neha Johnson"  
  }  
]
```

```
File Edit Selection View Go Run Terminal Help  
JS service.js  
1 const express = require('express');  
2 const app = express();  
3  
4 // Middleware to parse JSON  
5 app.use(express.json());  
6  
7 // Middleware to check version  
8 app.use('/api/users', (req, res, next) => {  
9   const version = req.headers['x-api-version'];  
10  if (!version) {  
11    return res.status(400).send('API version header is required');  
12  }  
13  req.version = version;  
14  next();  
15 });  
16  
17 // API versioning logic  
18 app.get('/api/users', (req, res) => {  
19   if (req.version === '1') {  
20     res.json([ { id: 1, name: 'Alice' }, { id: 2, name: 'Neha' } ]);  
21   } else if (req.version === '2') {  
22     res.json([ { id: 1, fullName: 'Alice Smith' }, { id: 2, fullName: 'Neha Johnson' } ]);  
23   } else {  
24     res.status(400).send('Unsupported API version');  
25   }  
26 });  
27  
28 app.listen(3000, () => console.log('Server is running on port 3000'));  
29  
30  
PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE CODE REFERENCE LOG TERMINAL  
[nodemon] starting 'node service.js'  
Server is running on port 3000  
API version requested: =2  
API version requested: 2  
API version requested: 1  
[nodemon] restarting due to changes...  
[nodemon] starting 'node service.js'  
Server is running on port 3000  
Ln 30, Col 1 Spaces: 4 UTF-8 CRLF JavaScript Signed out Go Live Quokka
```

Full Stack Web-D Interns L... Error Implement Versioning in Expre... Hoppscotch • Open source API API Header Versioning

hoppscotch.io

Try "Import collections" Ctrl K

Save My Workspace Login

HOPPSCOTCH

GET Untitled +

Select environment

Parameters Body Headers 2 Authorization Pre-request Script Tests Variables

Header List

Key	Value	Description	
x-api-version	1	Description	✓ ✖
x-api-version	2	Description	✓ ✖

Status: 200 • OK Time: 8 ms Size: 305 B

JSON Raw Headers 7 Test Results

Response Body

```
1 {
2   {
3     "id": 1,
4     "fullName": "Alice Smith"
5   },
6   {
7     "id": 2,
8     "fullName": "Neha Johnson"
9   }
10 }
```

Personal Workspace > Collections

Search

+ New

Collections are empty

Import or create a collection

Import

+ Add new

Help & feedback

Full Stack Web-D Interns L... Error Implement Versioning in Expre... Hoppscotch • Open source API

hoppscotch.io

Try "Share request" Ctrl K

Save My Workspace Login

HOPPSCOTCH

GET Untitled +

Select environment

Parameters Body Headers 1 Authorization Pre-request Script Tests Variables

Header List

Key	Value	Description	
x-api-version	1	Description	✓ ✖
x-api-version	2	Description	○ ✖

Status: 200 • OK Time: 5 ms Size: 281 B

JSON Raw Headers 7 Test Results

Response Body

```
1 {
2   {
3     "id": 1,
4     "name": "Alice"
5   },
6   {
7     "id": 2,
8     "name": "Neha"
9   }
10 }
```

Personal Workspace > Collections

Search

+ New

Collections are empty

Import or create a collection

Import

+ Add new

Help & feedback


```
File Edit Selection View Go Run Terminal Help ← → express-rest-api
Welcome JS service.js x
1 // server.js
2 const express = require('express');
3 const app = express();
4
5 // Middleware to parse JSON
6 app.use(express.json());
7
8 // API with query parameter versioning
9 app.get('/api/users', (req, res) => {
10   const version = req.query.version;
11   if (!version) {
12     return res.status(400).send('Version query parameter is required');
13   }
14
15   if (version === '1') {
16     res.json([ { id: 1, name: 'Alice' }, { id: 2, name: 'Neha' } ]);
17   } else if (version === '2') {
18     res.json([ { id: 1, fullName: 'Alice Smith' }, { id: 2, fullName: 'Neha Johnson' } ]);
19   } else {
20     res.status(400).send('Unsupported API version');
21   }
22 });
23
24 app.listen(3000, () => console.log('Server is running on port 3000'));
25
...
PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE CODE REFERENCE LOG TERMINAL
[nodeemon] starting 'node service.js'
Server is running on port 3000
[nodeemon] restarting due to changes...
[nodeemon] starting 'node service.js'
[nodeemon] clean exit - waiting for changes before restart
[nodeemon] restarting due to changes...
[nodeemon] starting 'node service.js'
Server is running on port 3000
Launchpad 0 0 0 Live Share Ln 25, Col 1 Spaces: 4 UTF-8 CRLF JavaScript Signed out Go Live Quokka
```

Full Stack Web-D Interns L... localhost:3000/api/users/version... Implement Versioning in Expre... Hoppscotch • Open source API

hoppscotch.io

HOPPSCOTCH Try "Share request" Save My Workspace Login

GET http://localhost:3000/api/users Send Save

Parameters Body Headers Authorization Pre-request Script Tests Variables

Query Parameters

version	1	Description	
version	2	Description	
Key	Value	Description	

Status: 200 • OK Time: 9 ms Size: 230 B

JSON Raw Headers Test Results

Response Body

```
1 {
2   {
3     "id": 1,
4     "name": "Alice"
5   },
6   {
7     "id": 2,
8     "name": "Neha"
9   }
10 }
```

Personal Workspace Collections

Search

+ New

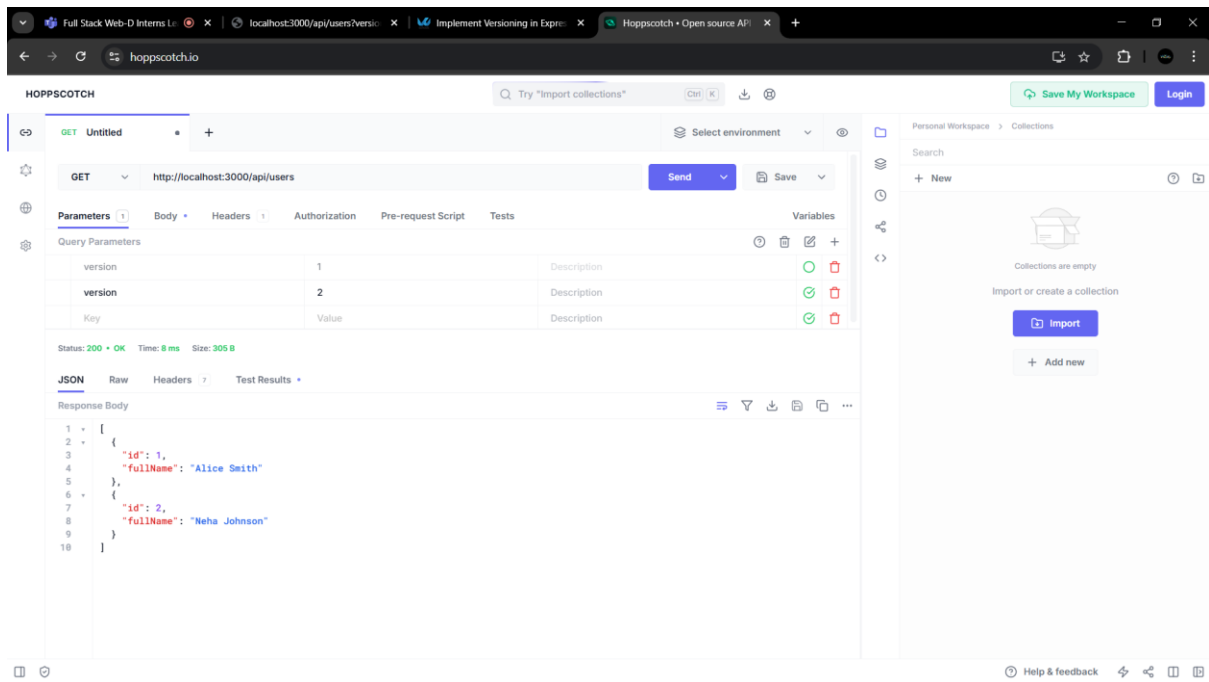
Collections are empty

Import or create a collection

Import

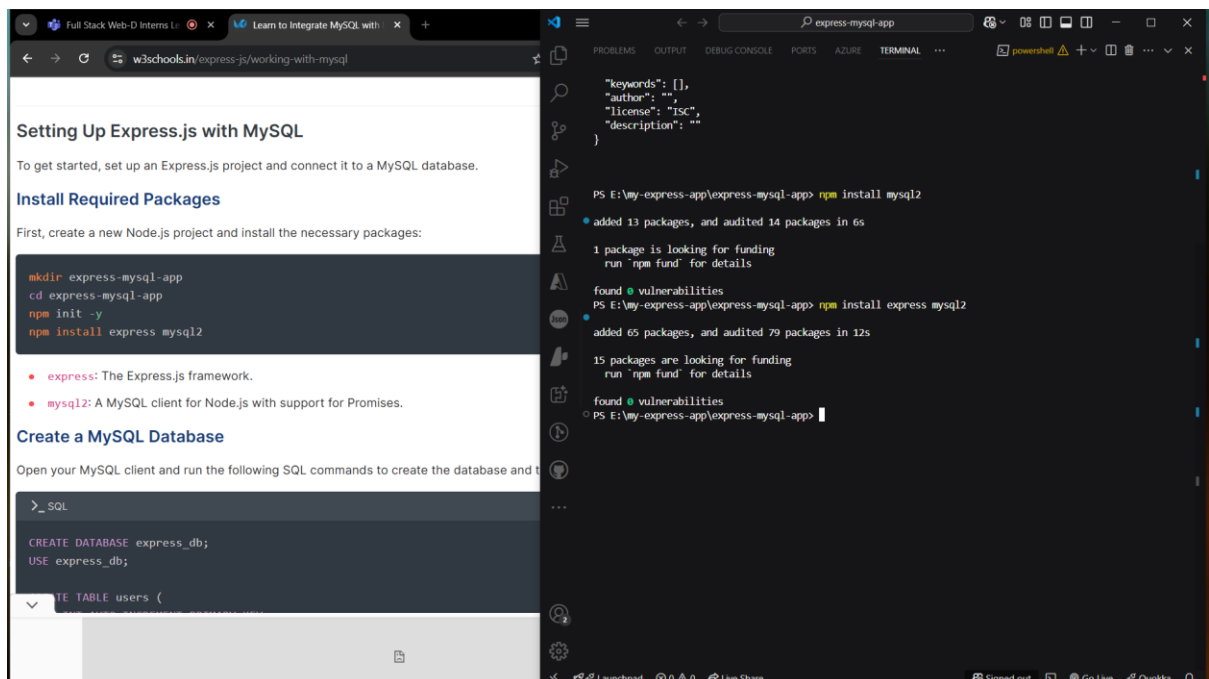
+ Add new

Help & feedback



SESSION 2(4:45pm – 6:30)

TOPIC:- Express.js My SQL



The screenshot shows a VS Code editor window titled 'express-mysql-app'. The Explorer sidebar on the left shows the project structure with files: db.js, app.js, node_modules, package-lock.json, package.json, and userModel.js. The main editor area displays the code in db.js, which is an async function connectDB() that attempts to connect to a MySQL database. The code includes a user object with 'root' as the username and 'Vaish@2005' as the password. It also includes a query to create a database named 'express_db' if it does not exist. The terminal at the bottom shows the command 'node db.js' being executed, which results in a 'Database connection failed: Unknown database 'express_db'' error. The status bar at the bottom indicates the file is at line 22, column 1, using UTF-8 encoding and CRLF line endings.

```
File Edit Selection View Go Run Terminal Help
express-mysql-app

EXPLORER
db.js
app.js
node_modules
package-lock.json
package.json
userModel.js

OPEN EDITORS
db.js
app.js

EXPRESS-MYSQL-APP
node_modules
package-lock.json
package.json
userModel.js

db.js
3 async function connectDB() {
4   const connection = await mysql.createConnection({
5     user: 'root',
6     password: 'Vaish@2005'
7   });
8
9   // Create DB if it doesn't exist
10  await connection.query("CREATE DATABASE IF NOT EXISTS express_db");
11  console.log("Database ensured");
12
13  // Switch to DB
14  await connection.changeUser({ database: "express_db" });
15  console.log("Connected to express_db");
16
17  return connection;
18 }
19
20 module.exports = connectDB;
21
22

PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE CODE REFERENCE LOG TERMINAL
node
at wrapModuleload (node:internal/modules/cjs/loader:235:24)
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:171:5)
at node:internal/main/run_main_module:36:49

Node.js v22.16.0
PS E:\my-express-app\express-mysql-app> node db.js
connected to MySQL!
PS E:\my-express-app\express-mysql-app> node app.js
Database connection failed: Unknown database 'express_db'
PS E:\my-express-app\express-mysql-app> node app.js
Database ensured
connected to express_db
Server running on port 3000

OUTLINE
TIMELINE
Ln 22, Col 1 Spaces: 4 UTF-8 CRLF JavaScript Signed out Go Live Quokka
```

CURD operation having some issue!
I will resolve and post it!