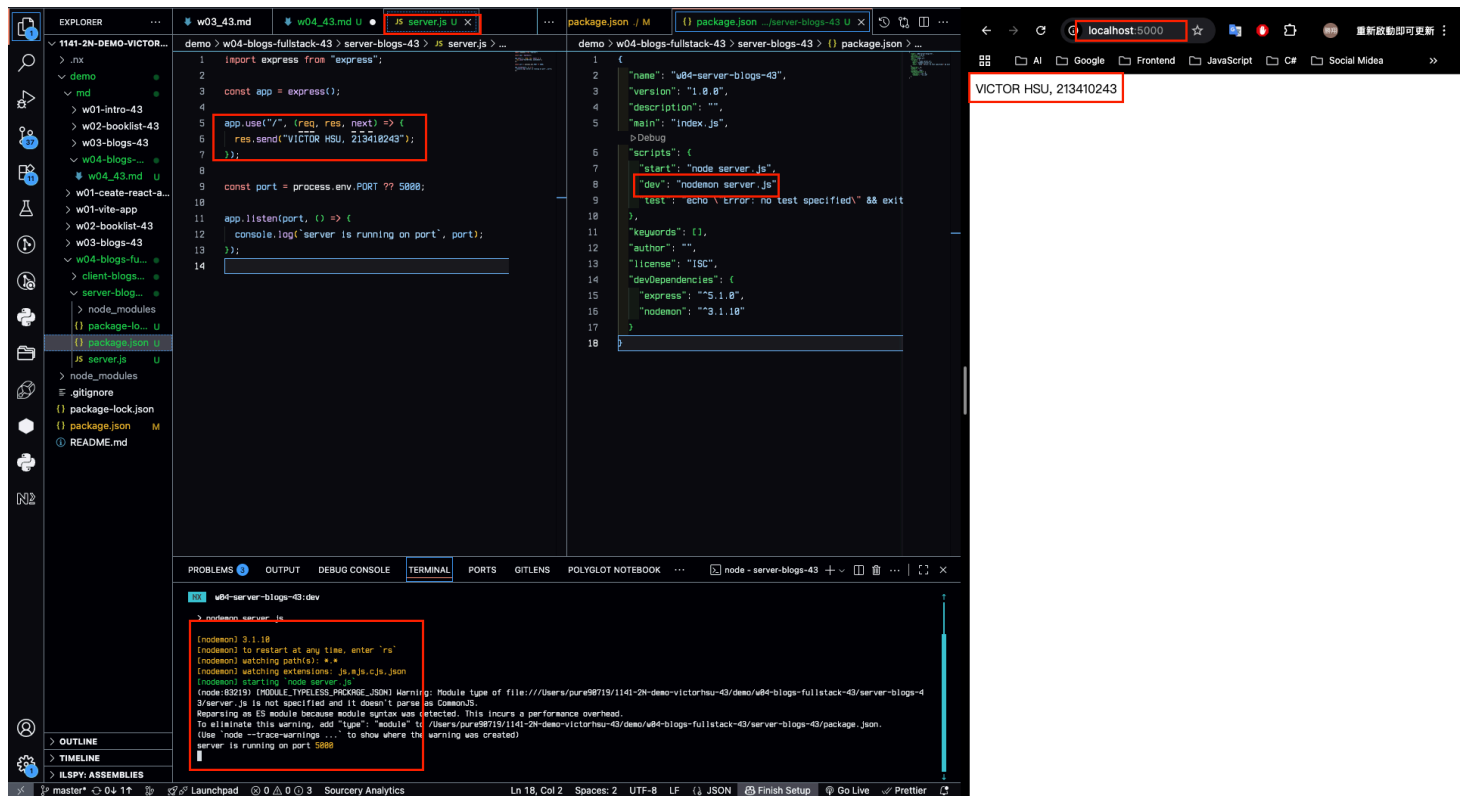


## W04-P1: Create a express Web server to show your info



778af01 victor\_xu

Sun Oct 12 12:29:03 2025 +0800 W04-P1: Create a express Web se

## W04-P2: Create blog\_43 table with 3 data, implement route /api/blog\_43 to return a json array with 3 data

=> SQL to create blog\_43 table and 3 data

The screenshot shows a PostgreSQL IDE interface. The left sidebar displays the 'Schemas (1)' tree with 'public' expanded, and 'Tables (1)' containing 'blog\_43'. The 'Columns' list for 'blog\_43' is shown, including 'id', 'title', 'descript', 'category', 'img', and 'remote\_img'. The main query editor contains the following SQL code:

```
1 CREATE TABLE blog_43 (  
2   id INT NOT NULL PRIMARY KEY,  
3   title VARCHAR(255),  
4   descript TEXT,  
5   category VARCHAR(255),  
6   img TEXT,  
7   remote_img TEXT  
8 );  
9  
10 INSERT INTO blog_43 (id, title, descript, category, img, remote_img)  
11 VALUES  
12 (1, 'Seven Reasons Why Coffee Is Awesome',  
13  'Lorem ipsum dolor sit amet consectetur adipisicing elit.',  
14  'lifestyle',  
15  '/images/photo-1.jpg',  
16  'https://erogcveccbzsyhbgputf.supabase.co/storage/v1/object/public/demo-xx/c  
17  
18 (2, 'Travel To Paris',  
19  'Lorem ipsum dolor sit amet consectetur adipisicing elit ';
```

The 'Data Output' tab shows the result: 'INSERT 0 3'. A message box at the bottom states: 'Query returned successfully in 33 msec.' The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.033' and 'LF Ln 28, Col 101'.

=> show 3 data

The screenshot shows the PostgreSQL 15 interface. On the left, the 'Schemas (1)' pane shows the 'public' schema containing a table named 'blog\_43'. The 'Columns' for 'blog\_43' are listed: id (integer, PK), title (character varying (255)), description (text), category (character varying (255)), img (text), and remote\_img (text). The 'Query' pane shows the following SQL query:

```
1 select * from public.blog_43
2 order by id asc
```

The 'Data Output' pane shows the results of the query, displaying 3 rows of data:

id	title	description	category	img	remote_img
1	Seven Reasons Why Coffee Is Awesome	Lorem ipsum dolor sit amet consectetur adipisicing elit.	lifestyle	/images/photo-1.jpg	https://erogvecbcszsyhbputf.supabase.co/storage/v1/object/public/demo-xx/card-xx/photo-1.jpg
2	Travel To Paris	Lorem ipsum dolor sit amet consectetur adipisicing elit.	travel	/images/photo-2.jpg	https://erogvecbcszsyhbputf.supabase.co/storage/v1/object/public/demo-xx/card-xx/photo-2.jpg
3	Coffee Brings Friendship	Lorem ipsum dolor sit amet consectetur adipisicing elit.	lifestyle	/images/photo-3.jpg	https://erogvecbcszsyhbputf.supabase.co/storage/v1/object/public/demo-xx/card-xx/photo-3.jpg

=> implement route /api/blog\_43

The screenshot shows the VS Code editor with the 'server-blogs-43' project. The 'w04\_43.js' file contains the following code:

```
1 import express from "express";
2 import db from "../database.js";
3
4 const app = express();
5
6 app.use("/api/blog_43", async (req, res, next) => {
7   const results = await db.query("select * from blog_43");
8   console.log("results", JSON.stringify(results));
9   res.json(results);
10 });
11
12 app.use("/", (req, res, next) => {
13   res.send("VICTOR HSU, 213418243");
14 });
15
16 const port = process.env.PORT ?? 5888;
17
18 app.listen(port, () => {
19   console.log("server is running on port", port);
20 });
```

The 'package.json' file shows the project dependencies:

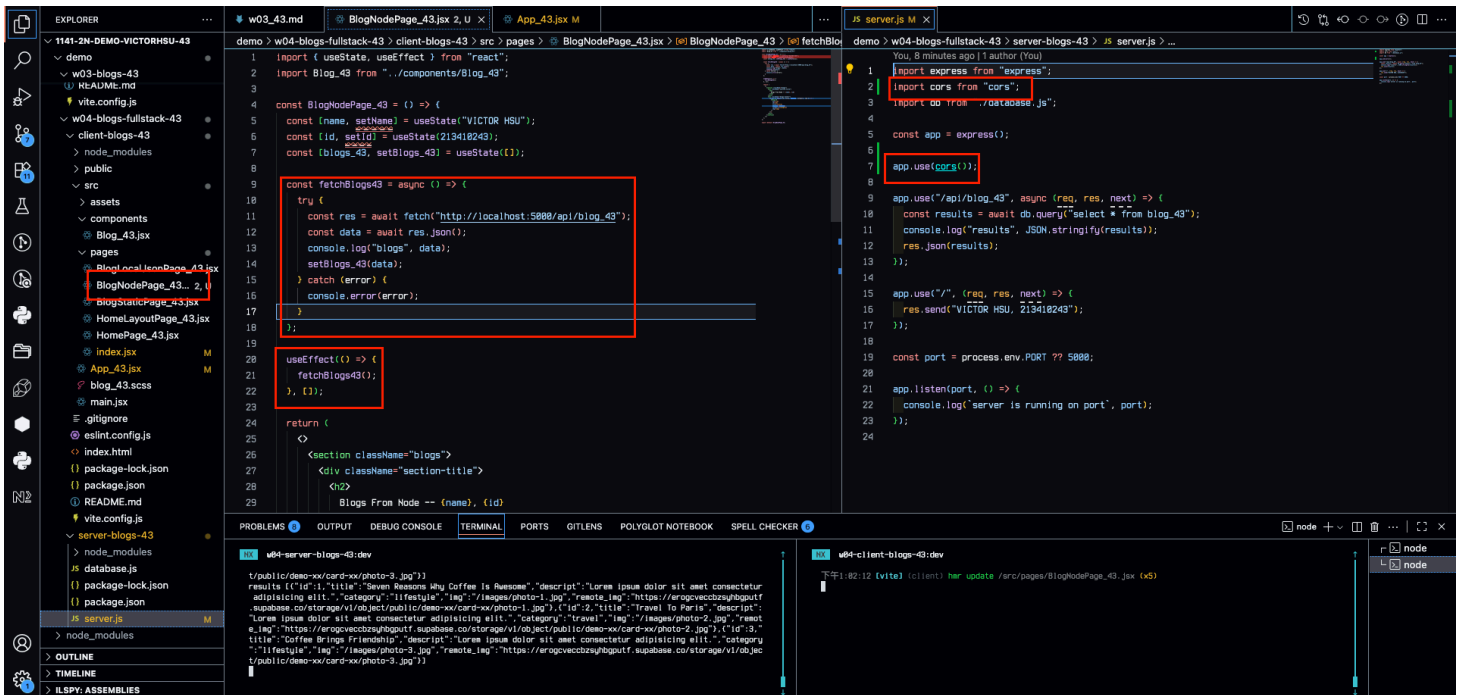
```
{
  "name": "w04-blogs-fullstack-43",
  "version": "1.0.0",
  "description": "w04-blogs-fullstack-43",
  "main": "index.js",
  "scripts": {
    "start": "node index.js",
    "dev": "nodemon index.js"
  },
  "dependencies": {
    "express": "^4.18.2",
    "pg": "^8.11.3",
    "pg-pool": "^3.6.1"
  },
  "devDependencies": {
    "nodemon": "^3.0.1"
  }
}
```

The 'index.js' file shows the server setup:

```
1 import express from "express";
2 import db from "../database.js";
3
4 const app = express();
5
6 app.use("/api/blog_43", async (req, res, next) => {
7   const results = await db.query("select * from blog_43");
8   console.log("results", JSON.stringify(results));
9   res.json(results);
10 });
11
12 app.use("/", (req, res, next) => {
13   res.send("VICTOR HSU, 213418243");
14 });
15
16 const port = process.env.PORT ?? 5888;
17
18 app.listen(port, () => {
19   console.log("server is running on port", port);
20 });
```

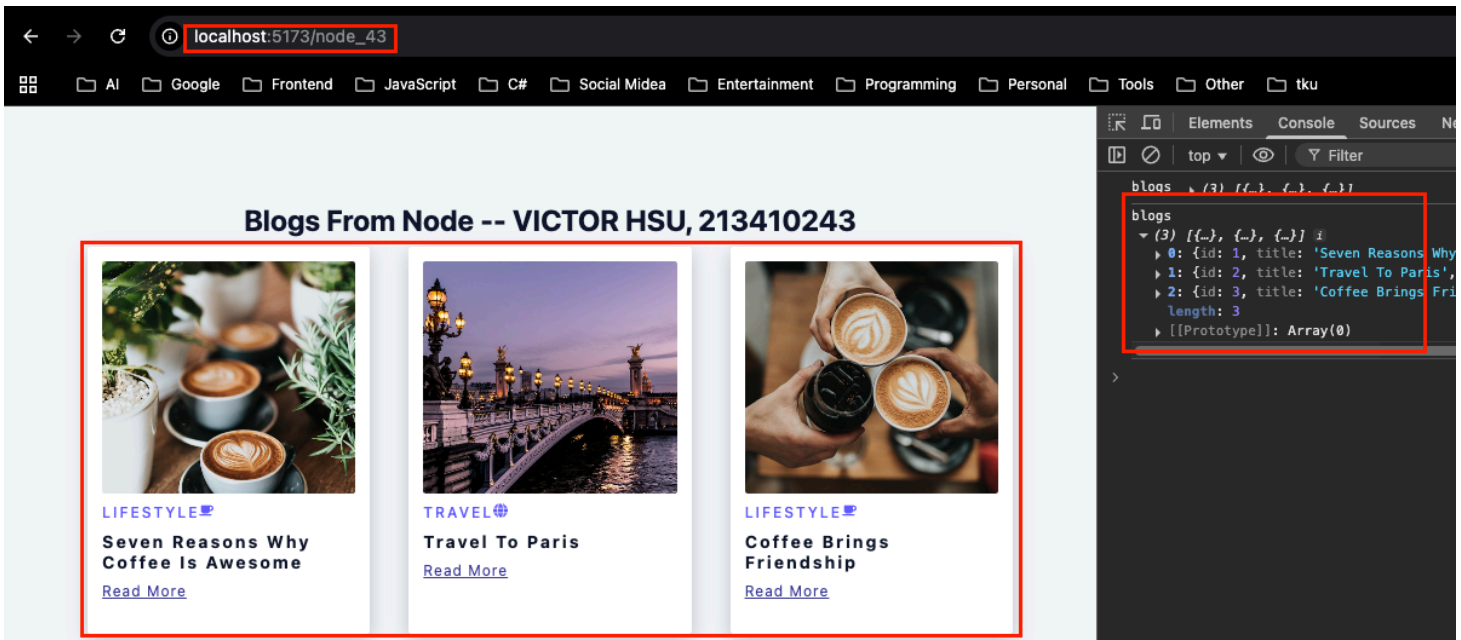
# W04-P3: from client side to get json data from Node

=> modified client and server code



The screenshot shows a VS Code editor with two files open: `BlogNodePage_43.jsx` and `server.js`. In `BlogNodePage_43.jsx`, the `fetchBlogs43` function is highlighted with a red box, showing it uses `fetch` to call `http://localhost:5888/api/blog_43`. The `useEffect` hook is also highlighted, showing it calls `fetchBlogs43()` when the component mounts. In `server.js`, the `cors` module is imported and `app.use(cors());` is added to the Express app. The `app.get('/api/blog_43')` route is also highlighted, showing it queries a database and returns the results as JSON.

=> Chrome, show 3 blogs



f097fc2 victor\_xu

Sun Oct 12 13:06:06 2025 +0800 W04-P3: from client side to get

# W04-logs: git logs of W04

vic0627 / 1141-2N-demo-victorhsu-43Public

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

Notifications

Commits

master

All usersAll time

Commits on Oct 12, 2025

W04-P3: from client side to get json data from Node

vic0627 committed 4 minutes ago

f0971c2

<>

W04-P2: Create blog\_43 table with 3 data, implement route /api/blog\_43 to return a json array with 3 data

vic0627 committed 16 minutes ago

94db9a3

<>

W04-P1: Create a express Web server to show your info

vic0627 committed 16 minutes ago

778af01

<>

Commits on Oct 3, 2025