

[Github URL](#)

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

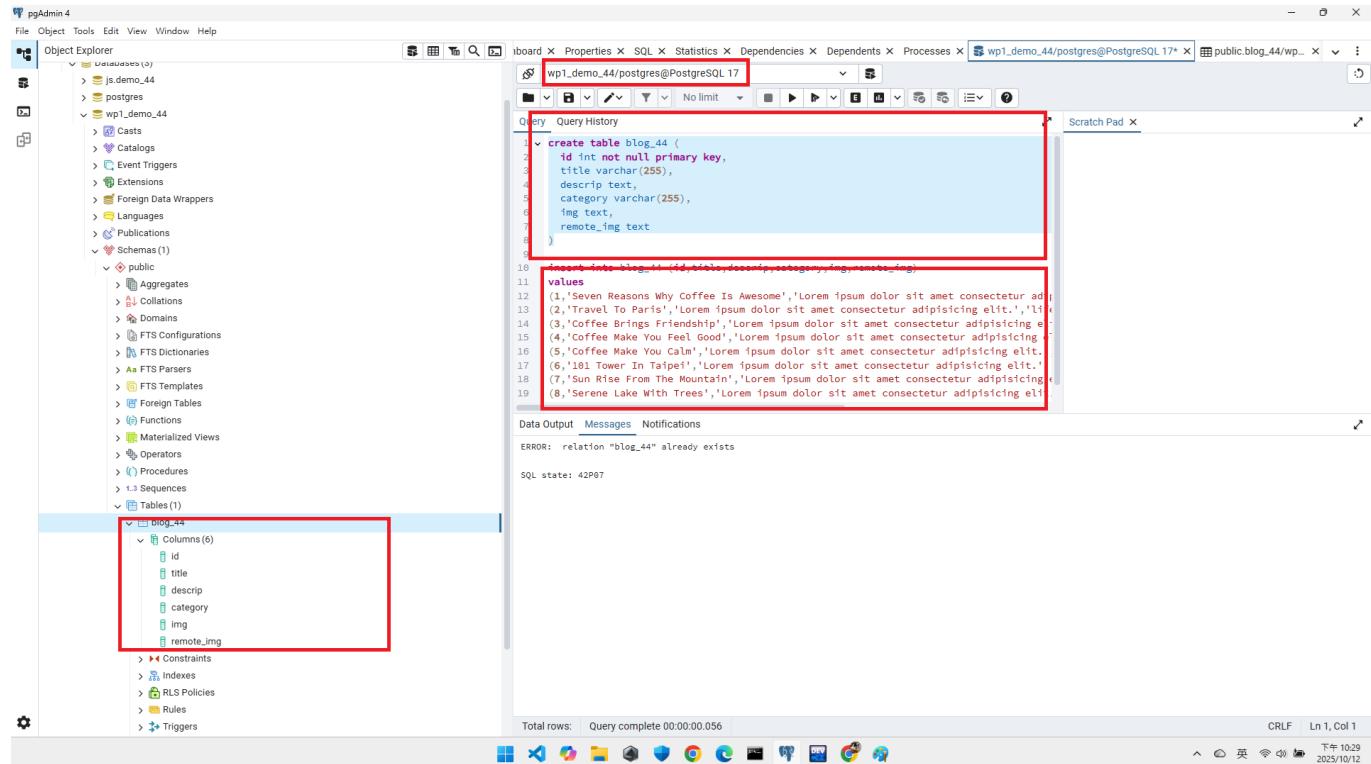
The screenshot shows the VS Code interface with the following components:

- Code Editor:** Two files are open: `server.js` and `package.json`. The `server.js` file contains code for an Express server that responds to the root route with the string "Hsu,213410144". The `package.json` file defines the project's metadata, including its name, version, description, main file, type, scripts (with "start" and "dev" keys), repository, keywords, author, license, and dependencies.
- Terminal:** Shows the command line output of running the application with `npm run dev`. It includes the command, the nodemon process starting, and the message "Server running on port 5000".
- Browser:** A Microsoft Edge browser window is open at `localhost:5000`, displaying the response from the server: "Hsu,213410144".
- Taskbar:** At the bottom, the taskbar shows various pinned icons and the date/time: "下午 09:06 2025/10/12".

```
3dae7ec 213410144      Sun Oct 12 21:12:31 2025 +0800 W04-P1: Create a express
Web server to show your info
```

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

=> **SQL to create blog_xx table and 3 data**



The screenshot shows the pgAdmin 4 interface. In the Object Explorer, under the 'wp1_demo_44' database, a new table named 'blog_44' is being created. The table has six columns: id, title, descrip, category, img, and remote_img. The 'id' column is defined as an integer with a primary key constraint. The 'title' column is a character varying (255) type. The 'descrip' column is a text type. The 'category' column is a character varying (255) type. The 'img' column is a text type. The 'remote_img' column is a text type.

```

create table blog_44 (
    id int not null primary key,
    title character varying(255),
    descrip text,
    category character varying(255),
    img text,
    remote_img text
)

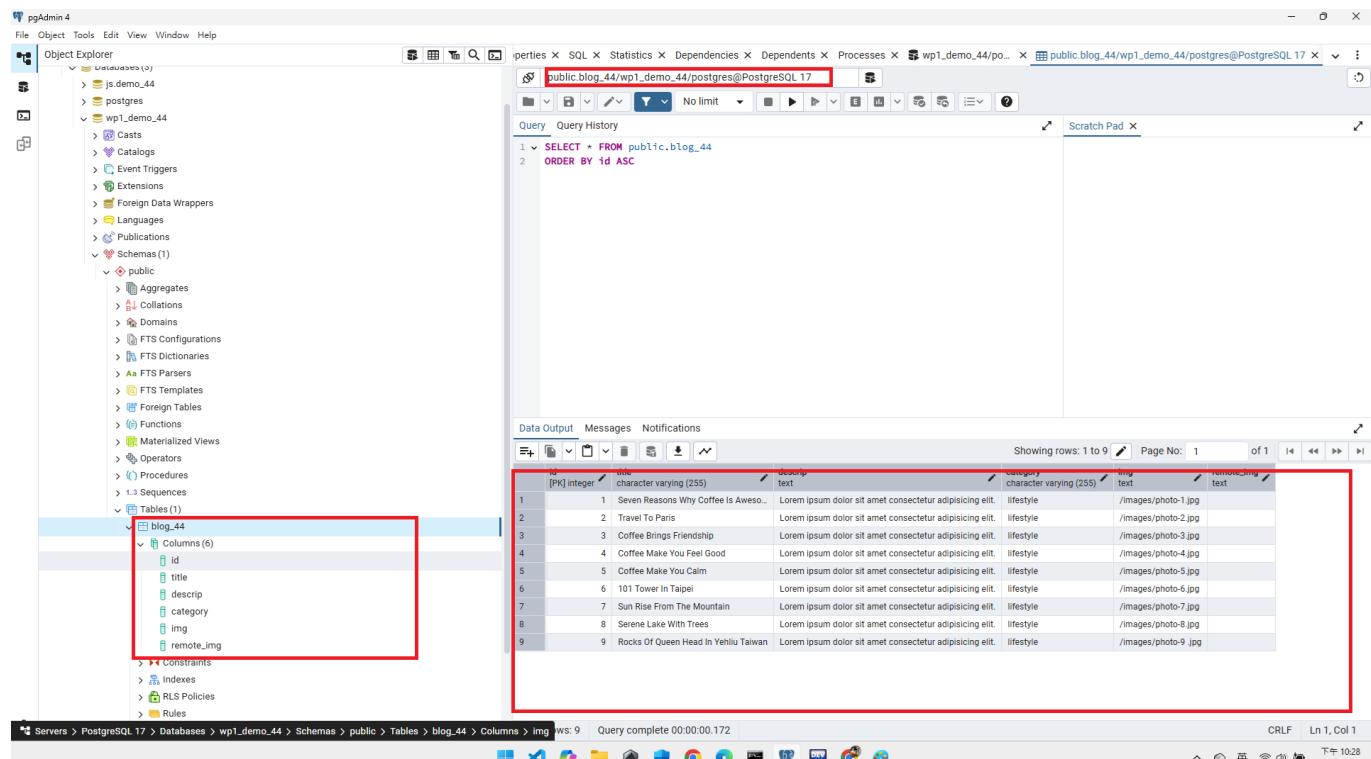
```

The 'values' section contains 9 rows of sample data:

- 1, 'Seven Reasons Why Coffee Is Awesome', 'Lorem ipsum dolor sit amet consectetur adipiscing elit.', 'lifestyle', '/images/photo-1.jpg'
- 2, 'Travel To Paris', 'Lorem ipsum dolor sit amet consectetur adipiscing elit.', 'lifestyle', '/images/photo-2.jpg'
- 3, 'Coffee Brings Friendship', 'Lorem ipsum dolor sit amet consectetur adipiscing elit.', 'lifestyle', '/images/photo-3.jpg'
- 4, 'Coffee Make You Feel Good', 'Lorem ipsum dolor sit amet consectetur adipiscing elit.', 'lifestyle', '/images/photo-4.jpg'
- 5, 'Coffee Make You Calm', 'Lorem ipsum dolor sit amet consectetur adipiscing elit.', 'lifestyle', '/images/photo-5.jpg'
- 6, '101 Tower In Taipei', 'Lorem ipsum dolor sit amet consectetur adipiscing elit.', 'lifestyle', '/images/photo-6.jpg'
- 7, 'Sun Rise From The Mountain', 'Lorem ipsum dolor sit amet consectetur adipiscing elit.', 'lifestyle', '/images/photo-7.jpg'
- 8, 'Serene Lake With Trees', 'Lorem ipsum dolor sit amet consectetur adipiscing elit.', 'lifestyle', '/images/photo-8.jpg'
- 9, 'Rocks Of Queen Head In Yehliu Taiwan', 'Lorem ipsum dolor sit amet consectetur adipiscing elit.', 'lifestyle', '/images/photo-9.jpg'

The SQL state is 42P07.

=> show 3 data



The screenshot shows the pgAdmin 4 interface. In the Object Explorer, under the 'wp1_demo_44' database, a query is run to select all columns from the 'blog_44' table and order them by 'id' in ascending order.

```

SELECT * FROM public.blog_44
ORDER BY id ASC

```

The results grid displays 9 rows of data, each corresponding to one of the 9 entries in the 'values' section of the previous screenshot. The columns are: id, title, descrip, category, img, and remote_img.

ID	Title	Description	Category	Image
1	Seven Reasons Why Coffee Is Awesome	Lorem ipsum dolor sit amet consectetur adipiscing elit.	lifestyle	/images/photo-1.jpg
2	Travel To Paris	Lorem ipsum dolor sit amet consectetur adipiscing elit.	lifestyle	/images/photo-2.jpg
3	Coffee Brings Friendship	Lorem ipsum dolor sit amet consectetur adipiscing elit.	lifestyle	/images/photo-3.jpg
4	Coffee Make You Feel Good	Lorem ipsum dolor sit amet consectetur adipiscing elit.	lifestyle	/images/photo-4.jpg
5	Coffee Make You Calm	Lorem ipsum dolor sit amet consectetur adipiscing elit.	lifestyle	/images/photo-5.jpg
6	101 Tower In Taipei	Lorem ipsum dolor sit amet consectetur adipiscing elit.	lifestyle	/images/photo-6.jpg
7	Sun Rise From The Mountain	Lorem ipsum dolor sit amet consectetur adipiscing elit.	lifestyle	/images/photo-7.jpg
8	Serene Lake With Trees	Lorem ipsum dolor sit amet consectetur adipiscing elit.	lifestyle	/images/photo-8.jpg
9	Rocks Of Queen Head In Yehliu Taiwan	Lorem ipsum dolor sit amet consectetur adipiscing elit.	lifestyle	/images/photo-9.jpg

=> implement route /api/blog_xx

The screenshot shows a developer's workspace with two main panes. The left pane is a code editor in Visual Studio Code (VS Code) displaying a file named `server.js`. The code implements a REST API endpoint for retrieving blog posts from a database. The right pane is a web browser window showing the results of the API call at `localhost:3000/api/blog_44`. The browser displays a JSON array of 10 blog post objects, each containing fields like `id`, `title`, `description`, `category`, `img`, and `remote_img`. A red box highlights the code in `server.js` and the resulting JSON data in the browser.

```
server.js
30
31 import db from './utils/database.js';
32
33 app.use(cors());
34
35 app.use('/api/blog_44', async (req,res, next) => {
36
37     const results = await db.query("select * from blog_44");
38
39     console.log("results", JSON.stringify(results.rows));
40
41     res.json(results.rows);
42
43 });
44
```

```
localhost:3000/api/blog_44
[
  {
    "id": 1,
    "title": "Seven Reasons Why Coffee Is Awesome",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit.",
    "category": "lifestyle",
    "img": "/images/photo-1.jpg",
    "remote_img": ""
  },
  {
    "id": 2,
    "title": "Travel To Paris",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit.",
    "category": "lifestyle",
    "img": "/images/photo-2.jpg",
    "remote_img": ""
  },
  {
    "id": 3,
    "title": "Coffee Brings Friendship",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit.",
    "category": "lifestyle",
    "img": "/images/photo-3.jpg",
    "remote_img": ""
  },
  {
    "id": 4,
    "title": "Coffee Make You Feel Good",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit.",
    "category": "lifestyle",
    "img": "/images/photo-4.jpg",
    "remote_img": ""
  },
  {
    "id": 5,
    "title": "Coffee Make You Calm",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit.",
    "category": "lifestyle",
    "img": "/images/photo-5.jpg",
    "remote_img": ""
  },
  {
    "id": 6,
    "title": "101 Tower In Taipei",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit.",
    "category": "lifestyle",
    "img": "/images/photo-6.jpg",
    "remote_img": ""
  },
  {
    "id": 7,
    "title": "Seven Reasons Why Sun Rise From The Mountain Is Amazing",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit.",
    "category": "lifestyle",
    "img": "/images/photo-7.jpg",
    "remote_img": ""
  },
  {
    "id": 8,
    "title": "Seren Lake With Trees",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit.",
    "category": "lifestyle",
    "img": "/images/photo-8.jpg",
    "remote_img": ""
  },
  {
    "id": 9,
    "title": "Rocks Of Queen Head In Yehliu Taiwan",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit.",
    "category": "lifestyle",
    "img": "/images/photo-9.jpg",
    "remote_img": ""
  },
  {
    "id": 10,
    "title": "Seven Reasons Why Coffee Is Awesome",
    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit.",
    "category": "lifestyle",
    "img": "/images/photo-10.jpg",
    "remote_img": ""
  }
]
```

3b16dc7 213410144

Tue Nov 11 22:45:36 2025 +0800 W04-P2: Create blog xx

table with 3 data, implement route /api/blog xx to return a json array with 3 data

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

=> modified client and server code

The screenshot shows a developer's workspace with multiple tabs open in a browser and several code editors.

OPEN EDITORS

- BlogNodePage_44.jsx (client_blog_44src)
- server.js (server_blog_44)
- BlogStatPage_44.jsx (client_blog_44src)
- index.jsx (client_blog_44src/pages)
- Homepage_44.jsx (client_blog_44src)
- HomeLayoutpage_44.jsx (client_blog_44src)
- blog_44.sql (server_blog_44utils/data)

GROUP 2

- W44
- client_blog_44
 - src
 - pages
 - BlogLocalizationPage_44.jsx
 - BlogNodePage_44.jsx
 - BlogStatPage_44.jsx
 - Homepage_44.jsx
 - index.jsx
 - App_44.jsx
 - App.css
 - blog_44.scss
 - index.css
 - main.jsx
 - .gitignore
 - eslint.config.js

OUTPUT

TERMINAL

```
User@UPTOP-BB159UBP MINGW64 /D/1141-2N-Hsu-44/2N-1/w04/client_blog_44 (main)
$ npx run dev
下午1:28:57 [vite] (client) Re-optimizing dependencies because vite config has changed

ROLLUP-VITE v7.1.12 ready in 476 ms

→ local: http://localhost:5173/
→ Network: use --host to expose
→ Network: use --https to enable https
下午1:31:18 [vite] (client) hmr update /src/pages/BlogNodePage_44.jsx
下午1:41:00 [vite] (client) hmr update /src/pages/BlogNodePage_44.jsx (x2)
下午1:41:38 [vite] (client) hmr update /src/pages/BlogNodePage_44.jsx (x3)
下午1:43:06 [vite] (client) hmr update /src/pages/BlogNodePage_44.jsx (x4)
```

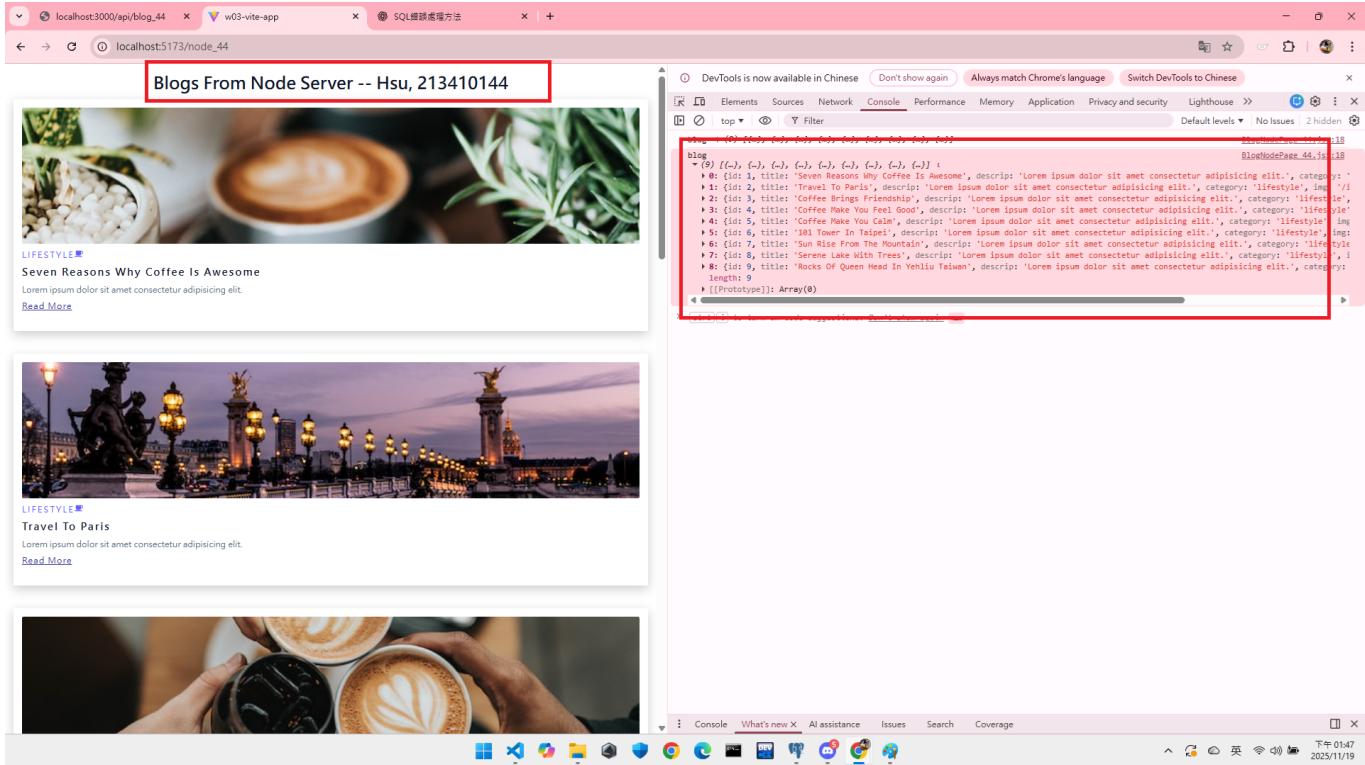
server_blog_44

```
server_blog_44.js
1 // server.js
2 // console.log(`Server running on port ${port}`);
3
4 import express from 'express';
5 import cors from 'cors';
6
7 const app = express();
8
9 app.use(cors());
10
11 app.use('/api/blog_44', async (req, res, next) => {
12   const results = await db.query(`select * from blog_44`);
13   res.json(results.rows);
14 })
15
16 app.get('/', (req, res) => {
17   res.send('Hsu,213418144');
18 })
19
20
21 const port = process.env.PORT || 3000;
22
23 app.listen(port, () => {
24   console.log(`Server running on port ${port}`);
25 })
```

POWER SHELL

- [powershell]
- [bash server_blog_44]
- [bash client_blog_44]

=> Chrome, show 3 blogs



5ab210c 213410144 Tue Nov 11 22:46:42 2025 +0800 W04-P3: from client side to get json data from Node

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

=> modified client and server code

```

client_blog_44.jsx
import { useState, useEffect } from 'react';
// Import blogData_44 from '../assets/data/blogData.json';
import Blog_44 from './components/Blog_44';

const api_url = 'http://localhost:3000/api/blog_44';

const BlogNodePage_44 = () => {
  const [name, setName] = useState('hsu');
  const [id, setId] = useState(213410144);

  // | 正確：預設值應為空陣列
  const [blog_44, setBlog_44] = useState([ ]);

  const fetchBlogFromNodeServer = async () => {
    try {
      const response = await fetch(api_url);
      const data = await response.json();
      console.log('blog', data);

      // | 正確：使用 setBlog_44
      setBlog_44(data);
    } catch (error) {
      console.log(error);
    }
  };

  useEffect(() => {
    fetchBlogFromNodeServer();
  }, [ ]);
}

export default BlogNodePage_44;
  
```

```

server_blog_44.js
const express = require('express');
import cors from 'cors';
const app = express();
import db from '../utils/database.js';

app.use(cors());
app.use('/api/blog_44', async (req, res, next) => {
  const results = await db.query(`select * from blog_44`);
  console.log('results', JSON.stringify(results.rows));
  res.json(results.rows);
});

app.use('/', (req, res, next) => {
  res.send('Hsu,213410144')
});

const port = process.env.PORT || 3000;
  
```

=> Chrome, show 3 blogs

The screenshot shows a web browser window with three tabs: 'localhost:3000/api/blog_44' (active), 'w03-vite-app', and 'SQL錯誤處理方法'. The main content area displays a blog post titled 'Blogs From Node Server -- Hsu, 213410144' with a red box highlighting it. Below the title is a large image of coffee and rosemary. The post is categorized under 'LIFESTYLE' and has a title 'Seven Reasons Why Coffee Is Awesome'. A snippet of the post content is shown: 'Lorem ipsum dolor sit amet consectetur adipisicing elit.'. There is a 'Read More' link. To the right of the browser is the Chrome DevTools 'Console' tab, which shows the JSON data returned from the API endpoint. The data is an array of 9 objects, each representing a blog post with fields like 'id', 'title', 'descrip', 'category', and 'img'. The DevTools interface includes tabs for Elements, Sources, Network, and Performance, along with a status bar at the bottom.

```
a94c50b 213410144      Wed Nov 19 19:01:27 2025 +0800 W04-P3: from client side  
to get json data from Node
```