

Github

Vercel

W13-P1: Get 3 menu data from /api/data_xx.json

The screenshot displays a development setup with two main windows. On the left is a code editor showing a file named `on.html` containing JavaScript code. A red box highlights the URL `127.0.0.1:5501/demo/w13-menu_31/menuLocalJson.html`. On the right is a web browser window showing the resulting page titled "Our Menu - Liang,208410331". The page lists three menu items: "Buttermilk Pancakes" (\$15.99), "Diner Double" (\$13.99), and "Godzilla Milkshake" (\$6.99). Each item has a small image and a brief description. A red box highlights the first menu item. Below the browser window is a developer tools panel showing the menu data structure. A red box highlights the first item in the array.

```
on.html
1 //多個進入需要加大括號，且menu名稱可以改
2 //import category from './bindata_31.js';
3 //import menu from './data_31.js';
4
5 //url可擺在local或遠端
6 const url = './api/data_31.json';
7 let menu = [];
8
9 //找到section-center的位置
10 const sectionCenter = document.querySelector('.section-center');
11 const btnContainer = document.querySelector('.btn-container');
12
13 const fetchData = async () => {
14   try {
15     const response = await fetch(url);
16     const data = response.json();
17     console.log('fetch menu', data);
18     return data;
19   } catch (error) {
20     console.log(error);
21   }
22 };
23
24 menu.map((item):轉成陣列把每一個item走一場
25 > const displayMenuItems = (menuItems) => {-
26   };
27
28 //回家作業答案-
29 const menuCategories = new Set(
30   menu.map(item) => {
31     return item.category;
32   }
33 );
34
35 //console.log('menuCategories', menuCategories);
36
37 const categories = ['all', ...menuCategories];
38 //console.log('categories', categories);
39
40 > const displayMenuButtons = () => {-
41   };
42
43 //dom跑進來後，呼叫一個程式；call back fn:() => {} ;資料抓回來之後通知
44 window.addEventListener('DOMContentLoaded', async () => {
45   menu = await fetchData();
46   displayMenuItems(menu);
47   displayMenuButtons();
48   //displayMenuButtons(category);
49 });
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
```

Our Menu - Liang,208410331

All

Buttermilk Pancakes \$15.99
Im baby woke milkshk wolf bitters live-edge blue bottle, hammock freegan copper mug whatever cold-pressed

Diner Double \$13.99
vaporware iPhone mumblecore selvage raw denim slow-carb leggings gochujang helvetica man braid jianbing. Marfa thundercats

Godzilla Milkshake \$6.99
ombucha chillwave fanny pack 3 wolf moon street art photo booth before they sold out organic viral.

fetch menu `menuLocalJson.js:17`
Promise {<pending>}
[[Prototype]]: Promise
[[PromiseState]]: "fulfilled"
[[PromiseResult]]: Array(3)
▶ 0: {id: 1, title: 'buttermilk pancakes'}
▶ 1: {id: 2, title: 'diner double', cate}
▶ 2: {id: 3, title: 'godzilla milkshake'}
length: 3
[[Prototype]]: Array(0)

filterBtns `menuLocalJson.js:83`
NodeList [button.filter-btn]

40a0b73 liangyu9103

Thu May 16 19:05:23 2024 +0800 ### W13-P1: Get 3 menu data from

W13-P2: Create 3 menu data from Supabase

=> create menu_xx table, and add 1 menu data

The screenshot shows the Supabase Table Editor interface. A new row is being created in the `menu_31` table. The row has the following data:

id	int8
1	2024-05-16 11:32:38.854432+08:00
title	buttermilk pancakes
category	breakfast
price	15.99
local_img	./images/item-1.jpeg
remote_img	https://sbwskgktsqzkhaohe.supabase.co/storage/public/demo/menu_31/item-1.jpeg
descrip	Im baby woke milshk wolf bitters live-edge blue hammock freedom copper mod whatever cold-dressed

The URL in the browser bar is `supabase.com/dashboard/project/sbwskgktsqzkhaohe/editor/29679`. The "Update row from" dropdown is set to `menu_31`. The "Optional Fields" section is visible, showing fields like `title`, `category`, `price`, `local_img`, `remote_img`, and `descrip`.

=> add RLS read policy for public access

The screenshot shows the Supabase Authentication Policies page. The left sidebar has a red box around the 'Authentication' section. Under 'CONFIGURATION', 'Policies' is selected and highlighted with a red box. A lock icon in the sidebar is also highlighted with a red box.

The main content area displays four tables:

- card_31**: Row Level Security disabled. A warning message states: "Warning: Row Level Security is disabled. Your table is publicly readable and writable. Anyone with the project's anonymous key can modify or delete your data. Enable RLS and create access policies to keep your data secure." Below it, a message says "No policies created yet".
- menu_31**: Row Level Security enabled. A green box highlights the "Row Level Security enabled" status. A red box highlights the policy definition: "SELECT * Enable read access for all users" and "Applied to: public".
- mid2_blog_31**: Row Level Security disabled. A warning message states: "Warning: Row Level Security is disabled. Your table is publicly readable and writable. Anyone with the project's anonymous key can modify or delete your data. Enable RLS and create access policies to keep your data secure." Below it, a message says "No policies created yet".
- product_31**: Row Level Security disabled. A warning message states: "Warning: Row Level Security is disabled. Your table is publicly readable and writable. Anyone with the project's anonymous key can modify or delete your data. Enable RLS and create access policies to keep your data secure." Below it, a message says "No policies created yet".
- task_31**: Row Level Security disabled. A red box highlights the "Row Level Security disabled" status.

=> add two more menu data from SQL command

We are currently investigating a technical issue, follow status.supabase.com for updates ↗

SQL Editor

liangyu9103's Org (Free) / tku_31 / ⚙ Enable branching Feedback 📱 ⓘ

Templates

Quickstarts

Search queries...

YOUR QUERIES

menu_31

mid2_blog_31

SQL for card_31

SQL for product_31

Table for Card Information

task_31

```
1 INSERT INTO menu_31 (id, title, category, price, local_img, remote_img, descrip)
2 values
3 (2, 'dinner double', 'dessert', 13.99, './images/item-2.jpeg',
4 'https://szbwskgktsqzkhaochez.supabase.co/storage/v1/object/public/
demo/menu_31/item-1.jpeg', 'vaporware iPhone mumblecore selvage raw
denim slow-carb leggings gochujang helvetica man braid jianbing.
Marfa thundercats'),
5 (3, 'godzilla milkshake', 'dinner', 6.99, './images/item-3.jpeg',
6 'https://szbwskgktsqzkhaochez.supabase.co/storage/v1/object/public/
demo/menu_31/item-2.jpeg', 'ombucha chillwave fanny pack 3 wolf moon
street art photo booth before they sold out organic viral.s')
```

Results Chart Export ⓘ role postgres Run ⌂

Success. No rows returned

=> show 3 menu data from menu_xx table

The screenshot displays two windows side-by-side. On the left is a code editor showing a JSON file named `data_31.json`. The JSON contains three menu items with IDs 1, 2, and 3, each with a title, category, price, local and remote image paths, and a detailed description. On the right is a Supabase Table Editor showing a table named `menu_31` with three rows corresponding to the menu items. The columns are `id`, `created_at`, `title`, and `category`.

Code Editor (Left):

```
[{"id": 1, "title": "buttermilk pancakes", "category": "breakfast", "price": 15.99, "local_img": "./images/item-1.jpeg", "remote_img": "", "descrip": "Im baby woke mlkshk wolf bitters live-edge blue bottle, hammock freegan copper"}, {"id": 2, "title": "diner double", "category": "dessert", "price": 13.99, "local_img": "./images/item-2.jpeg", "remote_img": "", "descrip": "vaporware iPhone mumblecore selvage raw denim slow-carb leggings gochujang h"}, {"id": 3, "title": "godzilla milkshake", "category": "dinner", "price": 6.99, "local_img": "./images/item-3.jpeg", "remote_img": "", "descrip": "ombucha chillwave fanny pack 3 wolf moon street art photo booth before they"}]
```

Table Editor (Right):

	id	created_at	title	category
1	1	2024-05-16 11:32:38.854432+00	buttermilk pancakes	breakfast
2	2	2024-05-16 11:47:19.026232+00	diner double	dessert
3	3	2024-05-16 11:47:19.026232+00	godzilla milkshake	dinner

ecc4990 liangyu9103 Thu May 16 19:58:07 2024 +0800 ### W13-P2: Create 3 menu data

W13-P3: Use menuSupa.html, menuSupa.js to get 3 menu data from Supabase

The screenshot shows a browser window displaying a menu page titled "Our Menu - Liang,208410331". The page lists three items: "Buttermilk Pancakes" (\$15.99), "Godzilla Milkshake" (\$6.99), and "Dinner Double" (\$13.99). Each item has a small image and a detailed description below it. Below the menu, the developer tools' Elements tab is open, showing the DOM structure of the menu items. The Network tab shows a fetch request to "menuSupa.js:17" which returns the menu data. The Sources tab shows the code for "menuSupa.js" and "menuSupa.html". The menuSupa.js file contains logic to fetch data from Supabase and render it into the DOM. The menuSupa.html file defines the structure of the menu items.

Our Menu - Liang,208410331

Buttermilk Pancakes \$15.99
Godzilla Milkshake \$6.99
Dinner Double \$13.99

```
fetch
  (3) [ {id: 1, created_at: '2024-05-16T11:32:38.854432+00:00', title: 'buttermilk pancakes', category: 'breakfast', price: 15.99}, {id: 3, created_at: '2024-05-16T11:47:19.026232+00:00', title: 'godzilla milkshake', category: 'dinner', price: 6.99}, {id: 2, created_at: '2024-05-16T11:47:19.026232+00:00', title: 'dinner double', category: 'dessert', price: 13.99} ]
```

```
const fetchData = async () => {
  const response = await fetch('https://api.supabase.com/v1beta1/databases/default/functions/public/menu');
  const data = await response.json();
  return data;
};

fetchData().then(data => {
  const menuItems = data.map(item => {
    const { id, title, category, price } = item;
    const remote_img = `https://res.cloudinary.com/.../image/upload/v1689409490/${item.id}.jpg`;
    return {
      id,
      title,
      category,
      price,
      remote_img
    };
  });
  const sectionCenter = document.querySelector('.section-center');
  sectionCenter.innerHTML = '';
  menuItems.forEach(item => {
    const { id, title, category, price, remote_img } = item;
    const menuItemElement = document.createElement('div');
    menuItemElement.classList.add('menu-item');
    menuItemElement.innerHTML = `
      
      

### ${title}



category: ${category}, price: ${price}



description: ${item.description}


    `;
    sectionCenter.appendChild(menuItemElement);
  });
});
```

menuSupa.html

menuSupa.js

clientSupabase_31.js

displayMenu

displayMenuItems

fetchData

menu

update

2024

menu

重新啟動即可更新

TKU iClass 1122 liang SQL Update Menu

127.0.0.1:5501/demo/w13-menu_31/menuSupa.html

GitHub Supabase Pictures | Unsplash W3Schools 線上成績平台 Project參考資料 所有書籤

All

元素 錄製工具 凸 主控台 原始碼 效能深入分析 凸 網路 效能 記憶體 應用程式 安全性 > 預設層級 沒有問題

filterBtns NodeList [button, filter-btn]

menuSupa.js:17

menuSupa.js:82

main

終端機

連接埠

輸出

偵錯主控台

zsh

問題

總共 14 (差異 5) · 復用 0 (差異 0), 重用包 0

remote: Resolving deltas: 100% (5/5), completed with 5 local objects.

To https://github.com/liangyu9103/1122-javascript

40a0b73..ecc4990 main --> main

(base) liangyu9103@liangyudelMacBook-Pro 1122-javascript % git log --pretty=format:"%h%x00%an%x00%ad%x00%as" --after="2024-05-15"

ecc4990 liangyu9103 Thu May 16 19:58:07 2024 +0800 ## W13-P2: Create 3 menu data from Supabase

40a0b73 liangyu9103 Thu May 16 19:05:23 2024 +0800 ## W13-P1: Get 3 menu data from /api/data_xx.json

(base) liangyu9103@liangyudelMacBook-Pro 1122-javascript %

Port: 5501 △ 10 Spell Prettier

1122-js-demo_31

```

menuSupa.html U > JS menuSupa.js U > w13_31.md M > JS clientSupabase_31.js U > JS menuSupa.js U >
demo > w13-menu_31 > menuSupa.html > html > body > script
2   <html lang="en">
3     <body>
4       <section class="menu">
5         <div class="section-center">
6           <ul>
7             <li>edison bulb yuccie crucifix microdosing.
8           </ul>
9         </div>
10        </article> -->
11      </div>
12    </section>
13
14    <script type="module"
15       src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js"
16     ></script>
17
18    <script type="module" src="./menuSupa.js"></script>
19
20  </body>
21 </html>
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

問題 62 緒論機 連接埠 輸出 儀錶主控台 POSTMAN CONSOLE

使用 8 個執行緒進行壓縮
壓縮物件中: 100% (12/12), 密度。
寫入檔案: 100% (14/14), 5.03 MiB | 3.45 MiB/s, 完成。
總共 141 (差異 5) - 徵用 0 (差異 0) / 廣用包 0
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/liangyu9103/1122-js-demo_31.git
 40a0b73..ecc4990 main -> main
 * [new branch] liangyu@chenliangyudeMacBook-Pro 1122-js-demo_31 % git log --pretty=format:"%h%x09%an%x09%ad%x09%" --after="2024-05-15"
 ecc4990 liangyu9103 Thu May 16 19:58:07 2024 +0800 ### W13-P2: Create 3 menu data from Supabase
 40a0b73 liangyu9103 Thu May 16 19:05:23 2024 +0800 ### W13-P1: Get 3 menu data from /api/data_xx.json
 * [base] liangyu@chenliangyudeMacBook-Pro 1122-js-demo_31 % []

第 188 行, 第 15 檞 空格: 2 UTF-8 LF HTML ⚡ Port: 5501 ⚡ 33 Spell ⚡ Prettier

b8b5e41 liangyu9103

Thu May 16 20:35:56 2024 +0800 ### W13-P3: Use menuSupa.html, |

W13-P4: Create company_xx and store_xx tables, and insert all data

=> data in store_xx table

The screenshot shows the Supabase Table Editor interface. On the left, there's a sidebar with various database objects like 'card_31', 'company_31' (which is highlighted with a red box), 'menu_31', etc. The main area is titled 'Table Editor' and shows the schema for the 'company_31' table:

	<code>id</code> int8	<code>created_at</code> timestamp	<code>name</code> varchar	<code>email</code> text	<code>phone</code> varchar
1		2024-05-16 13:26:41.630065	Ikea	ikea@gmail.com	(02)2120-4569
2		2024-05-16 13:26:41.630065	Liddy	liddy@gmail.com	(02)2351-4978
3		2024-05-16 13:26:41.630065	Marcos	marcos@gmail.com	(02)3198-9235
4		2024-05-16 13:26:41.630065	Caressa	caressa@gmail.com	(02)5288-2563

The 'Add RLS policy' button is also highlighted with a red box at the top right of the table view.

=> data in company_xx table

The screenshot shows the Supabase Table Editor interface. The left sidebar lists tables: card_31, company_31, menu_31, mid2_blog_31, product_31, products_31, store_31 (highlighted with a red box), and task_31. The main area displays the 'store_31' table with 12 rows. A red box highlights the 'Add RLS policy' button at the top right. Another red box highlights a tooltip for a foreign key relation: 'Foreign key relation: companyid → public.company_31.id'.

	id	created_at	title	price	com...	localimg	rem
1	2024-05-16 13:28:42.408038	Emperor Bed	21.99	1		./images/product-1.jpg	EMI
2	2024-05-16 13:28:42.408038	Accent Chair	25.99	2		./images/product-2.jpg	EMI
3	2024-05-16 13:28:42.408038	High-Back Bench	9.99	1		./images/product-3.jpg	EMI
4	2024-05-16 13:28:42.408038	Wooden Table	19.99	1		./images/product-4.jpg	EMI
5	2024-05-16 13:28:42.408038	Dining Table	6.99	4		./images/product-5.jpg	EMI
6	2024-05-16 13:28:42.408038	Entertainmint Center	21.99	2		./images/product-6.jpg	EMI
7	2024-05-16 13:28:42.408038	Albany Sectional	21.99	1		./images/product-7.jpg	EMI
8	2024-05-16 13:28:42.408038	Sofa Set	21.99	2		./images/product-8.jpg	EMI
9	2024-05-16 13:28:42.408038	Utopia Sofa	21.99	2		./images/product-9.jpg	EMI
10	2024-05-16 13:28:42.408038	Modern Bookshelf	21.99	3		./images/product-10.jpg	EMI
11	2024-05-16 13:28:42.408038	Albany Table	21.99	3		./images/product-11.jpg	EMI
12	2024-05-16 13:28:42.408038	Leather Sofa	21.99	2		./images/product-12.jpg	EMI

2ac1265 liangyu9103 Thu May 16 21:37:59 2024 +0800 ### W13-P4: Create company_xx a

W13 all code

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-05-15"
```

2ac1265 liangyu9103 Thu May 16 21:37:59 2024 +0800 ### W13-P4: Create company_xx a
b8b5e41 liangyu9103 Thu May 16 20:35:56 2024 +0800 ### W13-P3: Use menuSupa.html, i
ecc4990 liangyu9103 Thu May 16 19:58:07 2024 +0800 ### W13-P2: Create 3 menu data
40a0b73 liangyu9103 Thu May 16 19:05:23 2024 +0800 ### W13-P1: Get 3 menu data fro