

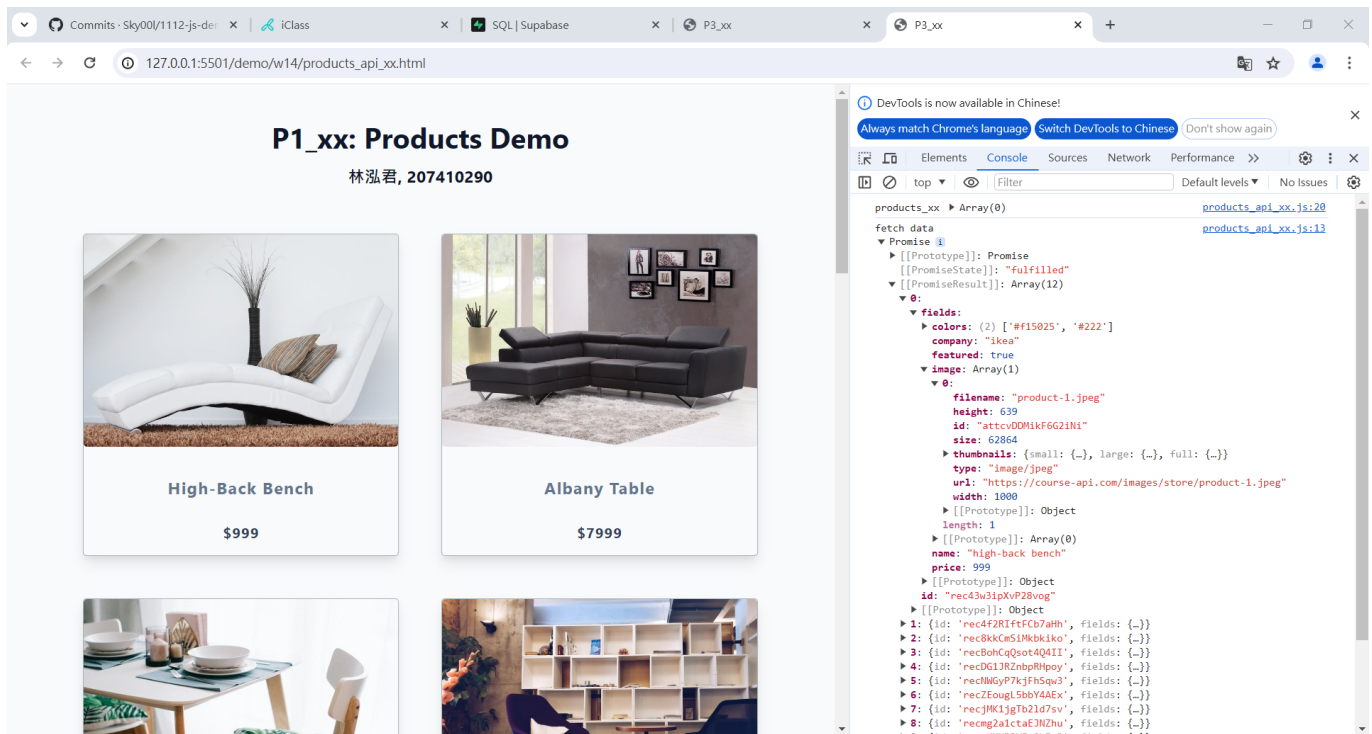
My Github URL

W14-P1: Do products_api_xx.js to get data from an api url

```

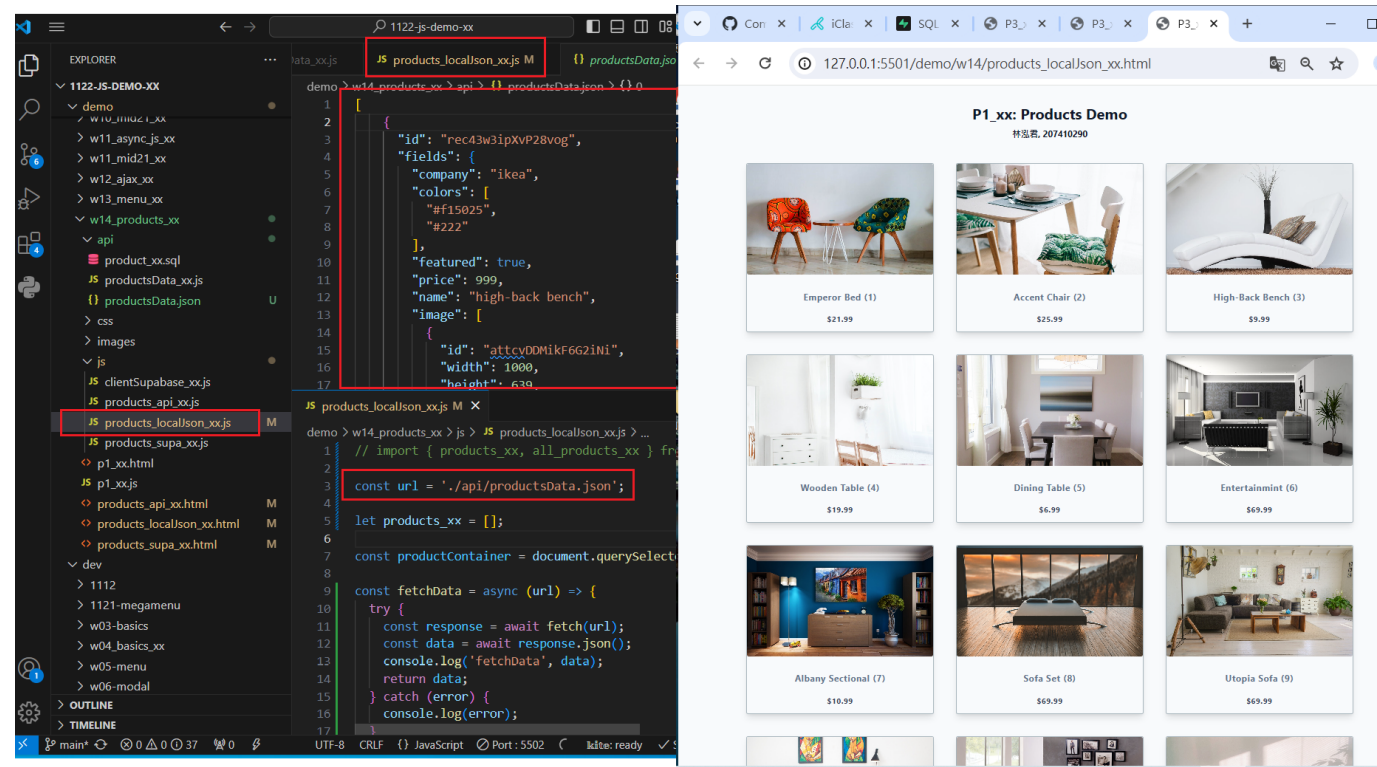
1 // import { products_xx, all_products_xx } from './p1_data_xx.js'
2
3 const url = 'https://www.course-api.com/javascript-store-products';
4
5 let products_xx = [];
6
7 const productContainer = document.querySelector('.products-container');
8
9 const fetchData = async (url) => {
10   try {
11     const response = await fetch(url);
12     const data = await response.json();
13     console.log('fetchData', data);
14     return data;
15   } catch (error) {
16     console.log(error);
17   }
18 };
19
20 console.log('products_xx', products_xx);
21
22 const displayProducts = (products) => {
23   let productsContent = products
24     .map((product) => {
25     const { name, price, image } = product.fields;
26     // const { id } = product;
27     return `
28       <div class="single-product">
29         
34         <footer>
35           <h3 class="name">${name}</h3>
36           <span class="price">${price}</span>
37         </footer>
38       </div>
39     `;
40   });
41   productContainer.innerHTML = productsContent;
42 };
43
44 document.addEventListener('DOMContentLoaded', async () => {
45   products_xx = await fetchData(url);
46   displayProducts(products_xx);
47 });

```



```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%" --after="2024-06-05"
0115e51 Sky001 Thu Feb 29 19:02:34 2024 +0800 W02-P1: Show Tictactoe css
```

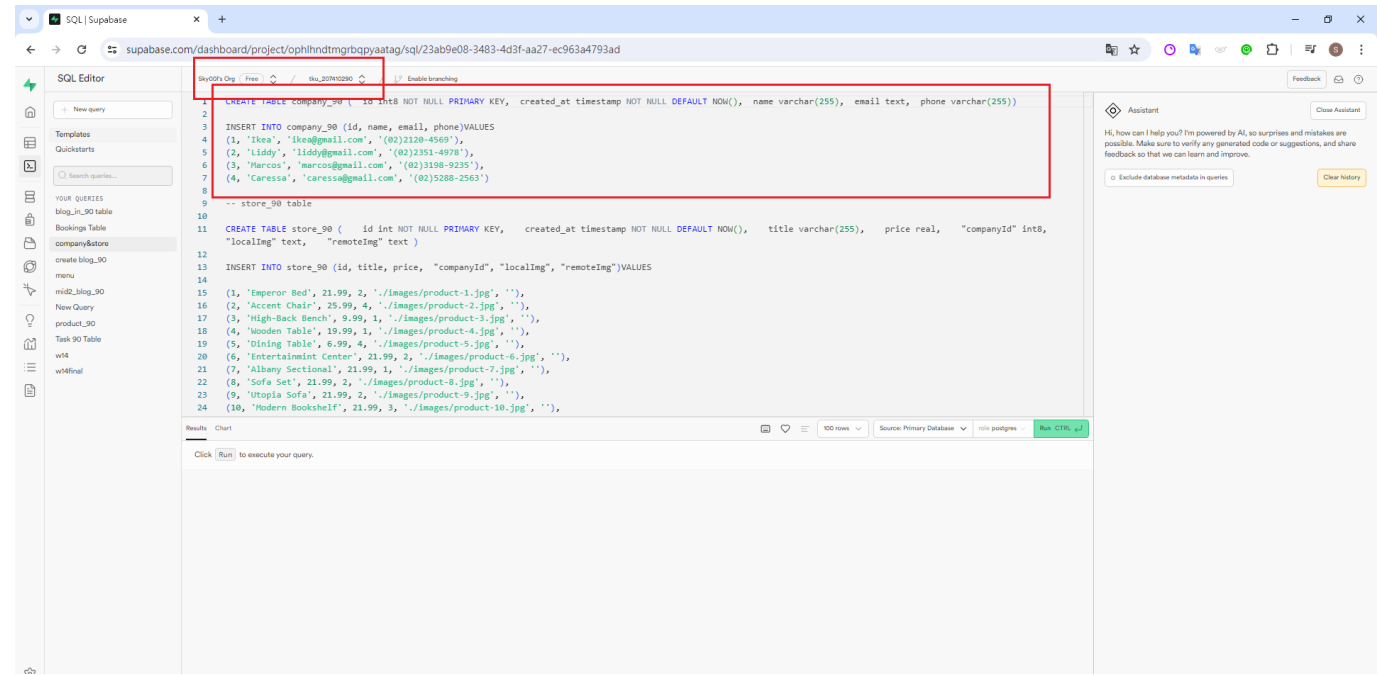
W14-P2: Do products_localJson_xx.js to get local json data (/api/productsData.json)



```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-05"
e9eae8d Sky001 Thu Jun 13 12:25:17 2024 +0800 W14-P2: Do
products_localJson_xx.js to get local json data (/api/productsData.json)
```

W14-P3: Use SQL to create schemas and data for company_xx and products_xx

=> company_xx schema and data



sky00's Org / Free / tku_207410290 / Enable branching

Filter | Sort | **Insert**

	id	created_at	name	email	phone
1	2024-06-06 02:54:19.153651	Ikea	ikea@gmail.com	(02)2120-4569	
2	2024-06-06 02:54:19.153651	Liddy	liddy@gmail.com	(02)3351-4978	
3	2024-06-06 02:54:19.153651	Marcos	marcos@gmail.com	(02)3198-9235	
4	2024-06-06 02:54:19.153651	Caressa	caressa@gmail.com	(02)5288-2563	

Page 1 of 1 | 100 rows | 4 records

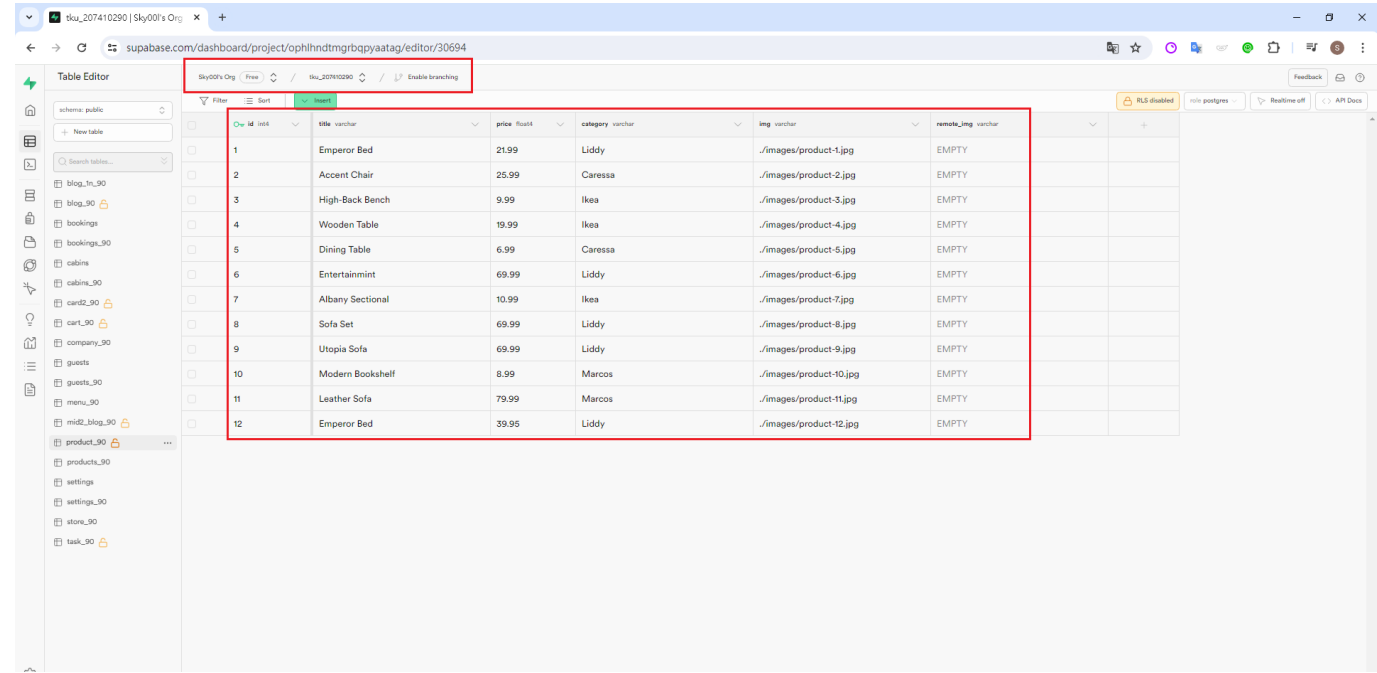
=> products_xx schema and data

SQL Editor

sky00's Org / Free / tku_207410290 / Enable branching

4 (1, 'Ikea', 'ikea@gmail.com', '(02)2120-4569'),
5 (2, 'Liddy', 'liddy@gmail.com', '(02)3351-4978'),
6 (3, 'Marcos', 'marcos@gmail.com', '(02)3198-9235'),
7 (4, 'Caressa', 'caressa@gmail.com', '(02)5288-2563')
8
9 -- store_90 table
10
11 CREATE TABLE store_90 (id int NOT NULL PRIMARY KEY, created_at timestamp NOT NULL DEFAULT NOW(), title varchar(255), price real, "companyId" int8,
12 "localImg" text, "remoteImg" text)
13
14 INSERT INTO store_90 (id, title, price, "companyId", "localImg", "remoteImg") VALUES
15
16 (1, 'Emperor Bed', 21.99, 2, './images/product-1.jpg', ''),
17 (2, 'Accent Chair', 25.99, 4, './images/product-2.jpg', ''),
18 (3, 'High-Back Bench', 9.99, 1, './images/product-3.jpg', ''),
19 (4, 'Wooden Table', 19.99, 1, './images/product-4.jpg', ''),
20 (5, 'Dining Table', 6.99, 4, './images/product-5.jpg', ''),
21 (6, 'Entertainment Center', 21.99, 2, './images/product-6.jpg', ''),
22 (7, 'Albany Sectional', 21.99, 1, './images/product-7.jpg', ''),
23 (8, 'Sofa Set', 21.99, 2, './images/product-8.jpg', ''),
24 (9, 'Utopia Sofa', 21.99, 2, './images/product-9.jpg', ''),
25 (10, 'Modern Bookshelf', 21.99, 3, './images/product-10.jpg', ''),
26 (11, 'Albany Table', 21.99, 3, './images/product-11.jpg', ''),
27 (12, 'Leather Sofa', 21.99, 2, './images/product-12.jpg', '')

Results | Chart | 100 rows | Source: Primary Database | role postgres | Run CTR: ⌘

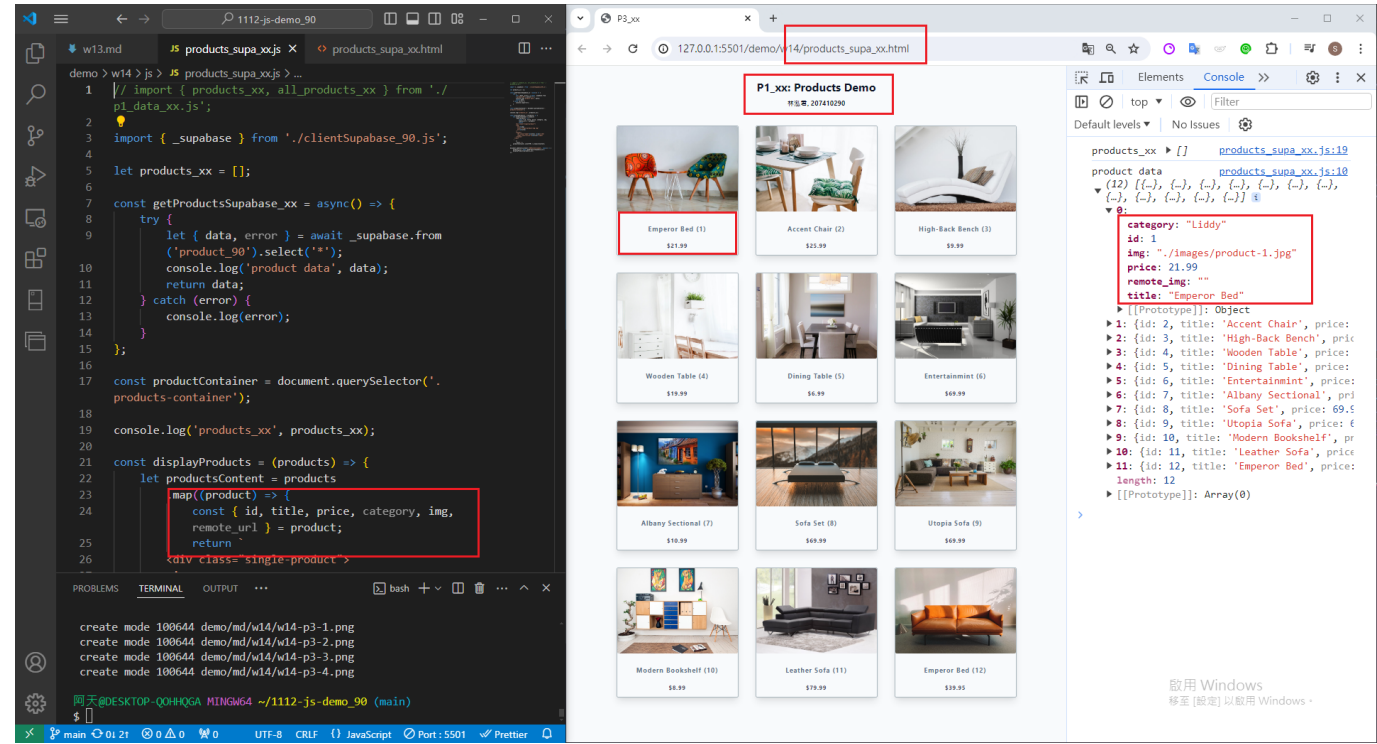


The screenshot shows the Supabase Table Editor interface. A table is being edited, and its data is as follows:

id	title	price	category	img	remote_img
1	Emperor Bed	21.99	Liddy	./images/product-1.jpg	EMPTY
2	Accent Chair	25.99	Caressa	./images/product-2.jpg	EMPTY
3	High-Back Bench	9.99	Ikea	./images/product-3.jpg	EMPTY
4	Wooden Table	19.99	Ikea	./images/product-4.jpg	EMPTY
5	Dining Table	6.99	Caressa	./images/product-5.jpg	EMPTY
6	Entertainment	69.99	Liddy	./images/product-6.jpg	EMPTY
7	Albany Sectional	10.99	Ikea	./images/product-7.jpg	EMPTY
8	Sofa Set	69.99	Liddy	./images/product-8.jpg	EMPTY
9	Utopia Sofa	69.99	Liddy	./images/product-9.jpg	EMPTY
10	Modern Bookshelf	8.99	Marcos	./images/product-10.jpg	EMPTY
11	Leather Sofa	79.99	Marcos	./images/product-11.jpg	EMPTY
12	Emperor Bed	39.95	Liddy	./images/product-12.jpg	EMPTY

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-05"
ad36c93 Sky001 Thu Jun 13 13:02:27 2024 +0800 W14-P3: Use SQL to create schemas
and data for company_xx and products_xx
```

W14-P4: Do products_supa_xx.js to get data from Supabase



The screenshot shows a development environment with three main components:

- Code Editor:** Displays the `products_supa_xx.js` file. The `getProductsSupabase_xx` function is defined, and the `displayProducts` function uses `map` to render the data. A red box highlights the `map` function's callback.
- Product Demo UI:** A grid of 12 product cards, each with an image, title, and price. The first card, 'Emperor Bed (1)' with a price of \$21.99, is highlighted with a red box.
- Console:** Shows the JSON data for the first product, 'Emperor Bed', with fields like `category`, `id`, `img`, `price`, and `remote_img`. A red box highlights the `category` and `id` fields.

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-05"
14f6388 Sky001 Thu Jun 13 13:11:27 2024 +0800 W14-P4: Do products_supa_xx.js to
get data from Supabase
```

W15-P5: Git logs for W15

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-05"
14f6388 Sky001 Thu Jun 13 13:11:27 2024 +0800 W14-P4: Do products_supa_xx.js to
get data from Supabase
ad36c93 Sky001 Thu Jun 13 13:02:27 2024 +0800 W14-P3: Use SQL to create schemas
and data for company_xx and products_xx
e9eae8d Sky001 Thu Jun 13 12:25:17 2024 +0800 W14-P2: Do
products_localJson_xx.js to get local json data (/api/productsData.json)
2fd4a8f Sky001 Thu Jun 13 12:24:28 2024 +0800 W14-P1: Do products_api_xx.js to
get data from an api url
755e18f Sky001 Thu Jun 13 12:03:23 2024 +0800 upload demo
```