

W3-P1: Use SupaClientGetBlog_xx.jsx (supabase client) to get card_xx data

The screenshot shows a development setup with a code editor and a browser window.

Code Editor (VS Code):

- File tree: W03-CRUD_31 project with files like SupaClientGetBlog_31.jsx, SupaGetBlog_31.jsx, App_31.jsx, etc.
- SupaClientGetBlog_31.jsx content (highlighted):


```

1 import { useState, useEffect } from 'react';
2 // let api url = http://localhost:5000/supabase/card_xx
3 import { supabase } from './db/clientSupabase_31';
4 const SupaClientGetBlog_31 = () => {
5   const [name, setName] = useState('Liang yu');
6   const [id, setId] = useState(208410331);
7   const [blogs, setBlogs] = useState([]);
8   const fetchBlogsFromSupabase = async () => {
9     try {
10       // const response = await fetch(api_url);
11       // const data = await response.json();
12       let { data, error } = await supabase.from('card_31')
13         .select('blogs_data', { data });
14       setBlogs(data);
15     } catch (error) {
16       console.log(error);
17     }
18   };
19   useEffect(() => {
20     fetchBlogsFromSupabase();
21   }, []);
22   return (
23     <section className="blogs">
24       <h2> Get Blogs Using Supabase Client </h2>
25       <h3> {` ${name}, ${id} `}</h3>
26       <div className="blogs-center">
27         <ul>
      
```

Browser Output:

- URL: localhost:5173
- Page Title: CRUD tutorial
- Content Area:
 - Get Blogs Using Supabase Client**
 - Liang yu, 208410331
 - Coffee photo
 - lifestyle
 - Seven Reasons Why Coffee Is Awesome**
 - Lorem ipsum dolor, sit amet consectetur adipisicing elit.
 - Travel To Paris**
 - Lorem ipsum dolor, sit amet consectetur adipisicing elit.
 - Seven Reasons Why Coffee Is Awesome**
 - Lorem ipsum dolor, sit amet consectetur adipisicing elit.
- Console Log (highlighted):


```

blogs data
(10) [{} , {} , {} , {} , {} ,
{} , {} , {} , {} , {} ]
0: {id: 1, title: 'Seven F
1: {id: 2, title: 'Travel
2: {id: 3, title: 'Seven F
3: {id: 4, title: 'Four Re
4: {id: 5, title: 'Five Re
5: {id: 6, title: 'Travel
6: {id: 7, title: 'Travel
7: {id: 8, title: 'Travel
8: {id: 9, title: 'Travel
9: {id: 10, title: 'Liang
length: 10
[[Prototype]]: Array(0)
      
```

ed74f0a 陳亮瑜 Wed Mar 6 20:02:41 2024 +0800 ### W3-P1: Use SupaClientGetBlog_xx.jsx

W3-P2: Use SupaGetBlog_xx.jsx to get card_xx data

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the file `SupaGetBlog_31.jsx` with several lines of code highlighted by a red box. The browser window shows the output of the code execution, displaying the title "CRUD tutorial" and a section titled "Delete Blogs Using http".

```
import { useState, useEffect } from 'react';
import axios from 'axios';

let apiUrl = 'https://szbwskgtsqzkhaochez.supabase.co/rest/v1';

const SupaClientGetBlog_31 = () => {
  const [name, setName] = useState('Liang Yu');
  const [id, setId] = useState(208410331);
  const [blogs, setBlogs] = useState([]);
  const fetchBlogsFromSupabase = async () => {
    try {
      const response = await axios.get(apiUrl, {
        headers: {
          apikey: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdX3hYAuthorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdX3hY',
        }
      });

      console.log('blogs data', response.data);
      // setBlogs(data);
    } catch (error) {
      console.log(error);
    }
  };
  useEffect(() => {
    fetchBlogsFromSupabase();
  }, []);
  return (
    <section className="blogs">
      <div>
        <h2> Get Blogs Using http </h2>
        <h3> {name}, {id} </h3>
      </div>
    </section>
  );
}

export default SupaClientGetBlog_31;
```

CRUD tutorial

Delete Blogs Using http

Liang Yu, 208410331

48d2ff4 陳亮瑜 Wed Mar 6 20:19:39 2024 +0800 ### W3-P2: Use SupaGetBlog_xx.jsx to get

W3-P3: Use SupaPostBlog_xx.jsx to add blog data

The screenshot shows a development environment with two tabs open: `SupaClientGetBlog_31.jsx` and `SupaPostBlog_31.jsx`. The `SupaPostBlog_31.jsx` tab is active, displaying a React component that performs an axios POST request to a Supabase endpoint. The code includes state management for name and ID, and handles the response from the database.

```
src > tutorials > SupaPostBlog_31.jsx > SupaClientGetBlog_31.jsx > fetchBlogsFromSupabase > response >
1 import { useState, useEffect } from 'react';
2 import axios from 'axios';
3
4 let apiUrl = "https://sbzbskgtksqzkhaohek.supabase.co/rest/v1/card_31";
5
6 const SupaClientGetBlog_31 = () => {
7   const [name, setName] = useState('Liang Yu');
8   const [id, setId] = useState(208410331);
9   const [blogs, setBlogs] = useState([]);
10  const fetchBlogsFromSupabase = async () => {
11    try {
12      const response = await axios.post(
13        apiUrl,
14        {
15          "id": 14,
16          "title": "Liang Yu 208410331",
17          "descrip": "Lorem ipsum dolor, sit amet consectetur adipisicing elit.",
18          "category": "travel",
19          "img": "/images/photo-1.jpg",
20        },
21      );
22
23      headers:
24      {
25        apikey:
26          'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3Mi0iJzdXbhYmFzZSIsInJlZiI6I
27          Authorization:
28            'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3Mi0iJzdXbhYmFzZSIsInJlZiI6I
29      }
30
31      console.log('blogs data', response);
32      //setBlogs(data);
33    } catch (error) {
34      console.log(error);
35    }
36  };
37}
38
39 useEffect(() => {
40   fetchBlogsFromSupabase();
41}, []);
```

The `POSTMAN` interface is also visible, showing a GET request to `https://sbzbskgtksqzkhaohek.supabase.co/rest/v1/card_31?select=*`. The response body contains the following JSON data:

```
{
  "id": 14,
  "title": "Liang Yu 208410331",
  "descrip": "Lorem ipsum dolor, sit amet consectetur adipisicing elit.",
  "category": "travel",
  "img": "/images/photo-1.jpg",
  "remote_url": null
}
```

38d339e 陳亮瑜 Wed Mar 6 20:23:04 2024 +0800 ### W3-P3: Use SupaPostBlog_xx.jsx to ad

W3-P4: Use SupaPatchBlog_xx.jsx to modify blog data

The screenshot shows two windows side-by-side. On the left is a code editor with the file `SupaPatchBlog_31.jsx` open. The code uses React hooks `useState` and `useEffect` along with `axios` to interact with a Supabase database. It patches a blog entry with ID 1, changing its title to "Liang Yu000". On the right is a web-based Table Editor from Supabase, showing a table named `card_31` with 13 rows of blog data. The columns are `id` (int4) and `title` (varchar). The data includes entries like "Seven Reasons Why Coffee Is Awesome", "Travel To Paris", and "Liang Yu 208410331".

```
import { useState, useEffect } from 'react';
import axios from 'axios';

let apiUrl = 'https://szbwskgtsqkhaokhez.supabase.co...';

const SupaPatchBlog_31 = () => {
  const [name, setName] = useState('Liang Yu');
  const [id, setId] = useState(208410331);
  const [blogs, setBlogs] = useState([]);

  const fetchBlogsFromSupabase = async () => {
    try {
      const response = await axios.patch(
        apiUrl,
        {
          title: 'Liang Yu000'
        },
        {
          headers: {
            'apikey': 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC...Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC...', ... });
        }
      );
      console.log('blogs data', response);
      //setBlogs(data);
    } catch (error) {
      console.log(error);
    }
  };
  useEffect(() => {
    fetchBlogsFromSupabase();
  }, []);
  return (
    ...
  );
}
```

	<code>id</code> int4	<code>title</code> varchar
1		Seven Reasons Why Coffee Is Awesome
2		Travel To Paris
3		Seven Reasons Why Coffee Is Awesome
4		Four Reasons Why Coffee Is Awesome
5		Five Reasons Why Coffee Is Awesome
6		Travel To Taipei
7		Travel To Alpha
8		Travel To Cairo
9		Travel To Taidong
12		Liang Yu 208410331
13		Liang Yu 208410331
14		Liang Yu 208410331
10		Liang Yu000

a042938 陳亮瑜 Wed Mar 6 20:49:31 2024 +0800 ### W3-P4: Use SupaPatchBlog_xx.jsx to m

W3-P5: Use SupaDeleteBlog_xx.jsx to delete blog data

The screenshot shows a code editor on the left and a Supabase Table Editor on the right. In the code editor, a file named `SupaDeleteBlog_31.jsx` is open, showing a React component that uses axios to delete a blog entry from a Supabase database. A red box highlights the line of code where the `delete` method is called on the `api_url`. In the Table Editor, there is a table named `card_31` with columns `id` and `title`. A red box highlights the rows with `id` values 13 and 10, which correspond to the entries that will be deleted.

```
src > tutorials > SupaDeleteBlog_31.jsx > SupaDeleteBlog_31 > fetchBlogsFromSupabase > response
1 import { useState, useEffect } from 'react';
2 import axios from 'axios';
3
4 let api_url = 'https://szbwskgtsqzkhaohek.supabase.co/rest/v1/card_31?id=<id>';
5
6 const SupaDeleteBlog_31 = () => {
7   const [name, setName] = useState('Liang Yu');
8   const [id, setId] = useState(208410331);
9   const [blogs, setBlogs] = useState([]);
10
11   const fetchBlogsFromSupabase = async () => {
12     try {
13       const response = await axios.delete(api_url,
14         {
15           headers: {
16             'apikey': 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3Mi0iJzdXBhYmFzZSIsInJlZii6I
17             Authorization: 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3Mi0iJzdXBhYmFzZSIsIn
18           },
19         }
20     );
21
22     console.log('blogs data', response);
23     //setBlogs(data);
24   } catch (error) {
25     console.log(error);
26   }
27 }
28
29
30 useEffect(() => {
31   fetchBlogsFromSupabase();
32 }, []);
33
34 return (
35   <section className="blogs">
36     <h2> Delete Blogs Using http </h2>
37     <h3>
38       <input type="text" value={name} onChange={(e) => setName(e.target.value)} />
39       <input type="button" value="Delete" onClick={() => setId(+name)} />
40     </h3>
41   </section>
42 )
43
44
```

id	title
1	Seven Reasons
2	Travel To Paris
3	Seven Reasons
4	Four Reasons V
5	Five Reasons W
6	Travel To Taipei
7	Travel To Alpha
8	Travel To Cairo
9	Travel To Taido
13	Liang Yu 20841
10	Liang Yu000

4373ba9 陳亮瑜 Wed Mar 6 20:51:07 2024 +0800 ### W3-P5: Use SupaDeleteBlog_xx.jsx to

W3-P6: Git log of W1

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-03-01"
```