

Your Github URL

Your Vercel URL

W12-P1: use xhr to display a simple text

The screenshot shows a code editor with an open file named `app_14.js`. The code uses XMLHttpRequest to fetch a file named `sample.txt` from a local directory. The browser window next to it displays the content of `sample.txt`, which is "30acf%09zero2005x%09Thu May 8 19:14:36 2025 +0800".

```

const xhr = new XMLHttpRequest();
console.log("xhr0", xhr);
xhr.open("GET", "./api/sample.txt");
console.log("xhr", xhr);
xhr.onreadystatechange = () => {
  if (xhr.readyState === 4 && xhr.status === 200) {
    const text = document.createElement('text');
    text.textContent = xhr.responseText;
    document.body.appendChild(text);
  } else {
    console.log({
      status: xhr.status,
      text: xhr.statusText,
    });
  }
};
xhr.send();

```

30acf%09zero2005x%09Thu May 8 19:14:36 2025 +0800 W12-P1: use xhr to display a simple text

W12-P2: click a button to fetch data

The screenshot shows a code editor with an open file named `app_14.js`. The code uses XMLHttpRequest to fetch a file named `sample.txt` from a local directory. The browser window next to it displays the content of `sample.txt`, which is "30acf%09zero2005x%09Thu May 8 19:14:36 2025 +0800".

```

const xhr = new XMLHttpRequest();
console.log("xhr0", xhr);
xhr.open("GET", "./api/sample.txt");
console.log("xhr", xhr);
xhr.onreadystatechange = () => {
  if (xhr.readyState === 4 && xhr.status === 200) {
    const text = document.createElement('text');
    text.textContent = xhr.responseText;
    document.body.appendChild(text);
  } else {
    console.log({
      status: xhr.status,
      text: xhr.statusText,
    });
  }
};
xhr.send();

```

1a194b8%09zero2005x%09Thu May 8 19:27:21 2025 +0800

W12-P2: click a button to fetch data

W12-P3: Run w10_product_supabase_xx, see how it works

=> `_supabase.from('product_xx').select('*');`

The screenshot shows a browser window with two tabs: 'w12.md M' and 'JS products_supabase_14.js'. The code in the JS tab is:

```

import { products_14 } from './products_data_14.js';
import { _supabase } from './clientSupabase_14.js';

const productContainer = document.querySelector('.products');

console.log("products_14", products_14);

const fetchProducts = async () => {
  try {
    const { data, error } = await _supabase.from("products")
      .select('*')
      .get();
    if (error) throw error;
    return data;
  } catch (error) {
    console.error("Error fetching products:", error);
    return [];
  }
};

const displayProducts = (products) => {
  let productsContent = products
    .map((product) => {
      const { id, title, price, img, remote_img } = product;
      return (
        

![{title}]({remote_img)

{title}



$21.99


      );
    });
  return productsContent;
};

```

The browser shows a product listing with two items:

- 1-Uncertain** \$21.99
- 2-Uncertain** \$25.99

The Network tab in the developer tools shows a single GET request to `https://npxswnlroydrfni.supabase.co/rest/v1/product_14?select=*` with a status of 200 OK.

=> check response

The screenshot shows a browser window with two tabs: 'w12.md M' and 'JS products_supabase_14.js'. The code in the JS tab is identical to the previous screenshot.

The browser shows a product listing with two items:

- 1-Uncertain** \$21.99
- 2-Uncertain** \$25.99

The Network tab in the developer tools shows a single GET request to `https://npxswnlroydrfni.supabase.co/rest/v1/product_14?select=*` with a status of 200 OK.

=> check how many http requests being done in fetchProducts

The screenshot shows a browser window with a Supaabase product demo. The page title is "Get Products Demo From Supaabase" by Liang-Ting Lin, 913410014. It displays two colorful chairs and a table. Below the image, it says "1-Uncertain" and "\$21.99". The Network tab in the developer tools shows various file requests, including "product-1.jpg" through "product-12.jpg", "product-6.jpg" (failed), "powerShell", "product.supta_14.html", "supabase.js", "product-14.css", "products_supta_14.js", "products_data_14.js", and "clientSupabase_14.js". The total transfer time is 48.10s.

cbf093a%09zero2005x%09Thu May 8 20:01:34 2025 +0800
w10_product_supabase_xx, see how it works

W12-P3: Run

W12-P4: Fetch person.json string and display name in the browser

=> use **JSON.parse()** to convert **responseText** to **JSON array**

The screenshot shows a browser window with an "Asynchronous Javascript - AJAX Demo" by Itlin, 913410014. It features a "click me" button. The Network tab in the developer tools shows a request for "person.json" which returns a JSON array of names:

```
[{"id":1,"name":"htchung"}, {"id":2,"name":"htchung2"}, {"id":3,"name":"htchung3"}, {"id":4,"name":"htchung4"}, {"id":5,"name":"htchung5"}, {"id":6,"name":"htchung6"}, {"id":7,"name":"htchung7"}, {"id":8,"name":"htchung8"}, {"id":9,"name":"htchung9"}]
```

=> extract name from data and show it in the browser

```
nc_14 > 3-fetch-json > JS app_14.js > [object Object] > onreadystatechange > [object Object] > displayData > [object Object] > data.map() callback
43    };
10    console.log("xhr0", xhr);
11
12    xhr.open("GET", url);
13    console.log("xhr", xhr);
14
15    xhr.onreadystatechange = () => {
16        console.log("xhr", xhr);
17        if (xhr.readyState === 4 && xhr.status === 200) {
18            const data = JSON.parse(xhr.responseText);
19            console.log("data", data);
20            // console.log("data in string", JSON.stringify(data));
21
22            const displayData = data
23                .map((item) => {
24                    const { id, name } = item;
25                    // return `<p>${id} : ${name}</p>`;
26                    return `<p>${name}</p>`;
27                })
28                .join("");
29                console.log("displayData", displayData);
30                const createElement = document.createElement("div");
31                createElement.classList.add("data");
32                createElement.innerHTML = displayData;
33                document.body.appendChild(createElement);
34            } else {
35                console.log({
36                    status: xhr.status,
37                    text: xhr.statusText,
38                });
39            }
40        };
41
42        xhr.send();

```

iClass Document 127.0.0.1:50733/demo/w12_ajax_async_14/3-f...

Elements Console Sources Network Filter Default levels No Issues

Hide network Log XMLHttpRequests

Preserve log Eager evaluation

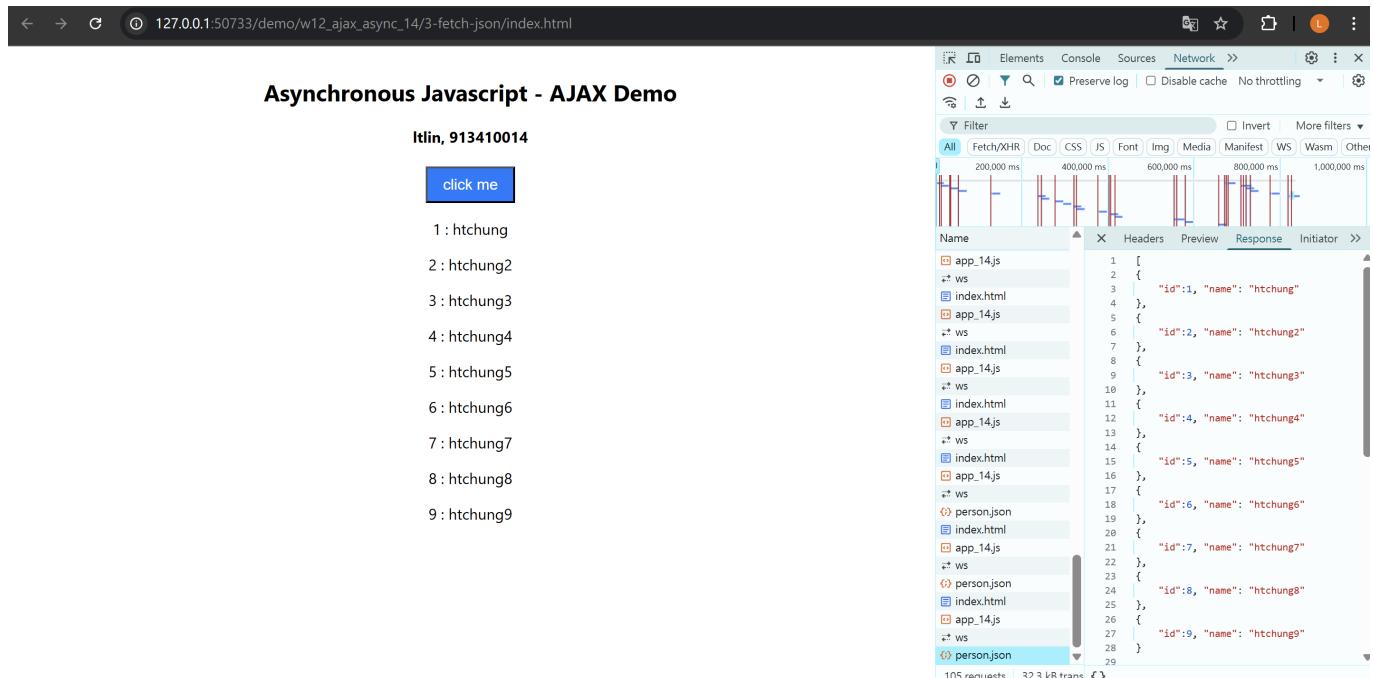
Selected context only Autocomplete from history

Group similar messages in console Treat code evaluation as user action

Show CORS errors in console

```
> t: 0, withCredentials: false, upload: XMLHttpRequestUpload, ...}
xhr
  XMLHttpRequest {onreadystatechange: null, readyState: 1, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, ...}
> t: 0, withCredentials: false, upload: XMLHttpRequestUpload, ...
xhr
  XMLHttpRequest {readyState: 2, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, onreadystatechange: f, ...}
> {status: 200, text: 'OK'}
  XMLHttpRequest {readyState: 3, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, onreadystatechange: f, ...}
> {status: 200, text: 'OK'}
  XMLHttpRequest {readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, onreadystatechange: f, ...}
data
  (9) [{}]
    > 0: {id: 1, name: 'htchung'}
    > 1: {id: 2, name: 'htchung2'}
    > 2: {id: 3, name: 'htchung3'}
    > 3: {id: 4, name: 'htchung4'}
    > 4: {id: 5, name: 'htchung5'}
    > 5: {id: 6, name: 'htchung6'}
    > 6: {id: 7, name: 'htchung7'}
    > 7: {id: 8, name: 'htchung8'}
    > 8: {id: 9, name: 'htchung9'}
      length: 9
      [Prototype]: Array(0)
displayData <htchung1></p><htchung2></p>
<htchung3></p><htchung4></p><htchung5></p><htchung6></p>
<htchung7></p><htchung8></p><htchung9></p>
```

=> check the Network, http response



fa81c08%09zero2005x%09Thu May 8 20:42:58 2025 +0800
string and display name in the browser

W12-P4: Fetch person.json

W11-log: qit logs for W12

The screenshot shows a GitHub repository page for 'zero2005x / 1132-IN-DEMO-LTLIN-14'. The 'Commits' tab is selected. A dropdown menu shows 'main' is selected. Filter buttons for 'All users' and 'All time' are visible. The commit list is organized by date:

- Commits on May 8, 2025**
 - W12-P4: Fetch person.json string and display name in the browser**
zero2005x committed 1 minute ago · ✓ 1 / 1
 - W12-P3: Run w10_product_supra_xx, see how it works**
zero2005x committed 43 minutes ago · ✓ 1 / 1
 - W12-P2: click a button to fetch data**
zero2005x committed 1 hour ago · ✓ 1 / 1
 - W12-P1: use xhr to display a simple text**
zero2005x committed 1 hour ago · ✓ 1 / 1
- Commits on May 1, 2025**
 - W11-P5: Change Github repo name, and Vercel URL**
zero2005x committed last week · ✓ 1 / 1
 - W11-P4: Use Asvnc Await to W11-P3**
—