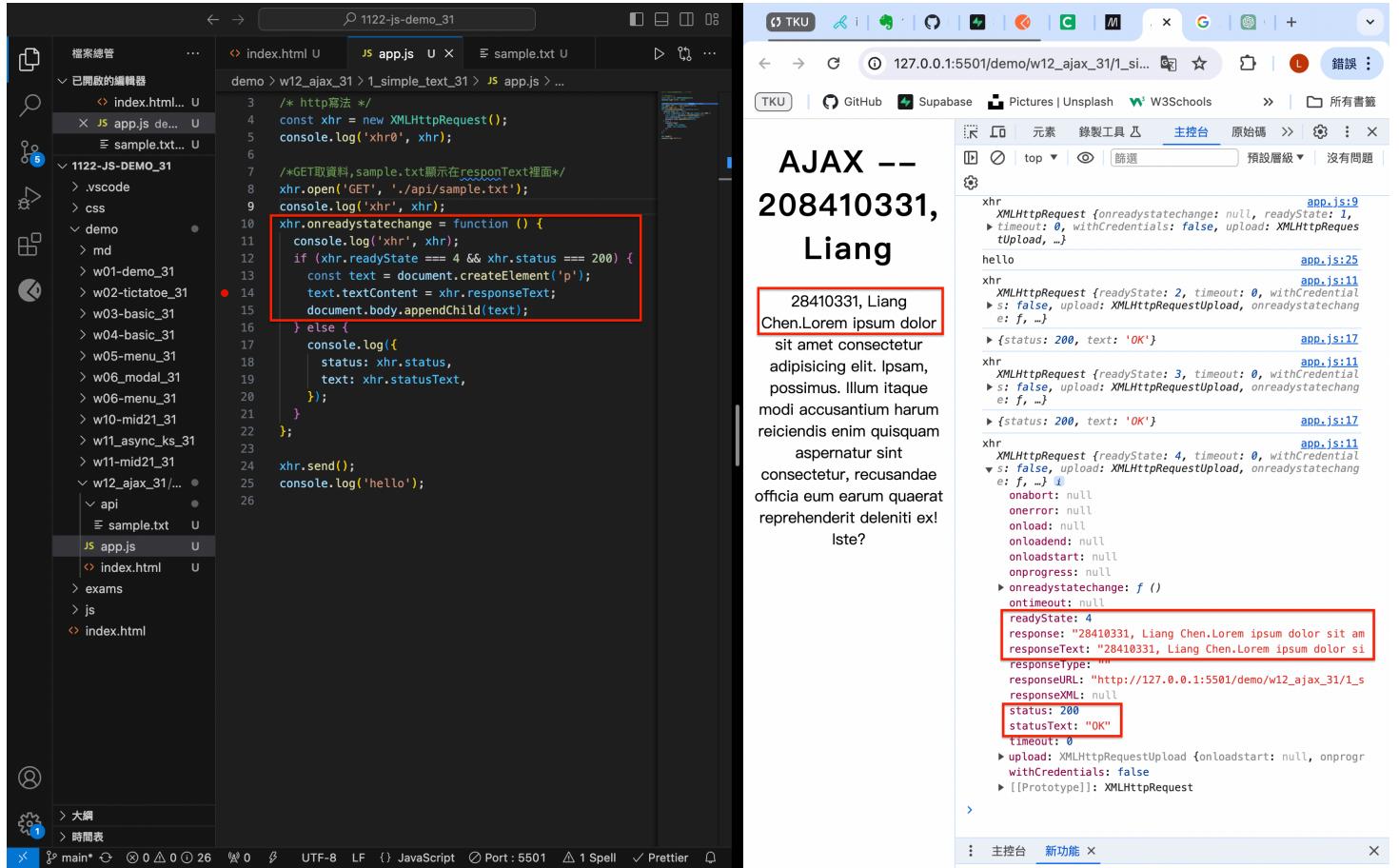


Github

Vercel

W12-P1: Do 1_Sample-text



The screenshot shows a browser developer tools console with two tabs: '主控台' (Console) and '新功能' (New Features). The '主控台' tab displays the following XMLHttpRequest logs:

```
XMLHttpRequest {readyState: 1, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, ...}
hello
app.js:25
XMLHttpRequest {readyState: 2, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, onreadystatechange: f, ...}
{x: "28410331, Liang Chen.Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsam, possimus. Illum itaque modi accusantium harum reiciendis enim quisquam aspernatur sint consectetur, recusandae officia eum earum querat reprehenderit deleniti ex! Iste?"}
app.js:17
XMLHttpRequest {readyState: 3, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, onreadystatechange: f, ...}
{x: "28410331, Liang Chen.Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsam, possimus. Illum itaque modi accusantium harum reiciendis enim quisquam aspernatur sint consectetur, recusandae officia eum earum querat reprehenderit deleniti ex! Iste?"}
app.js:11
XMLHttpRequest {readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, onreadystatechange: f, ...}
{x: "28410331, Liang Chen.Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsam, possimus. Illum itaque modi accusantium harum reiciendis enim quisquam aspernatur sint consectetur, recusandae officia eum earum querat reprehenderit deleniti ex! Iste?"}
```

The '樣式' (Style) tab on the left shows a preview of the CSS rule `.app.js {background-color: #f0f0f0; color: #000;}`.

493e73e liangyu9103

Thu May 9 19:17:21 2024 +0800 ### W12-P1: Do 1_Sample-text

W12-P2: Do 2_add_btn

=> Network, Header

The screenshot shows a browser developer tools interface with several tabs open. The Network tab is active, displaying a list of network requests. One request, labeled 'app.js', is highlighted with a red box. The details panel for this request shows the following information:

- Request URL: http://127.0.0.1:5501/demo/w12_ajax_31/2_add_btn/app.js
- Method: GET
- Status: 304 Not Modified
- Response Headers:
 - Content-Type: application/javascript
 - Cache-Control: no-cache, no-store, must-revalidate, max-age=0, s-maxage=0
 - ETag: W/"12-0.0.1:5501"
 - Last-Modified: Thu May 9 19:17:21 2024 +0800
 - Expires: -1
 - Date: Thu, 09 May 2024 19:17:21 GMT
 - Content-Length: 0
- Request Headers:
 - Accept: */*
 - Accept-Encoding: gzip, deflate
 - Accept-Language: zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7
 - Cache-Control: no-cache
 - Host: 127.0.0.1:5501
 - Origin: http://127.0.0.1:5501
 - Referer: http://127.0.0.1:5501/demo/w12_ajax_31/2_add_btn/app.js
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.195 Safari/537.36

=> Network, Response

The screenshot shows a browser developer tools window with two main tabs: "Network" and "Response".

Network Tab: This tab displays a list of network requests. One request, "index.html", is highlighted. Its details are shown in the bottom pane:

- Request Headers:** Content-Type: application/javascript, Accept: */*, Referer: https://127.0.0.1:5501/demo/w12_aj..., User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.135 Safari/537.36, Host: 127.0.0.1:5501.
- Request Body:** (empty)
- Response Headers:** Content-Type: text/html; charset=UTF-8, Content-Length: 100, Date: Mon, 07 May 2024 10:46:00 GMT, Vary: Accept-Encoding, X-Powered-By: PHP/8.2.10, X-Content-Type-Options: nosniff, X-Frame-Options: SAMEORIGIN, X-XSS-Protection: 1; mode=block, Cache-Control: no-store, no-cache, must-revalidate, max-age=0, Pragma: no-cache.
- Response Body:** (content of index.html page)

Response Tab: This tab displays the content of the "index.html" response. It includes a "show text" button and the following text:

28410331, Liang Chen. Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsam, possimus. Illum itaque modi accusantium harum reiciendis enim quisquam aspernatur sint consectetur, recusandae officia eum earum quaerat reprehenderit deleniti ex! Iste?

The "Network" tab also lists other requests such as "app.js", "ws", and "sample.txt".

53cc848 liangyu9103 Thu May 9 19:46:00 2024 +0800 ### W12-P2: Do 2_add_btn

W12-P3: Do 3_json_people_xhr

The screenshot shows a code editor and a browser developer tools interface. In the code editor, the file `app.js` contains JavaScript code for handling an XMLHttpRequest to a JSON file. A red box highlights the section where the XMLHttpRequest is opened and the response is processed. In the browser's Network tab, a request to `people.json` is shown, and the response body is displayed as a JSON array of four objects.

```
const btn = document.querySelector('.btn');
const url = './api/people.json';

btn.addEventListener('click', () => {
  getData(url);
});

function getData() {
  const xhr = new XMLHttpRequest();
  console.log('xhr', xhr);
  console.log('GET資料, 傳到url裡面');
  xhr.open('GET', url);
  console.log('xhr', xhr);
  xhr.onreadystatechange = function () {
    if (xhr.readyState === 4 && xhr.status === 200) {
      console.log('xhr', xhr);
      const data = JSON.parse(xhr.responseText);
      const displayData = data
        .map((item) => {
          return `<p>${item.name}</p>`;
        })
        .join('');
      /*產生一個div把資料塞在裡面 */
      const element = document.createElement('div');
      element.innerHTML = displayData;
      document.body.appendChild(element);
    } else {
      console.log({
        status: xhr.status,
        text: xhr.statusText,
      });
    }
  };
  xhr.send();
}

console.log('hello');
```

```
[{"id": 1, "name": "Liang"}, {"id": 2, "name": "208410331"}, {"id": 3, "name": "AAA"}, {"id": 4, "name": "BBB"}]
```

4269bb0 liangyu9103

Thu May 9 20:19:54 2024 +0800

W12-P3: Do 3_json_people_xhr

W12-P4: Do 3_json_people_fetch_api

=> `fetch(url).then(response => response.json())`

The screenshot shows a development setup with multiple windows:

- Left Window (VS Code):** Displays the file structure of a project named "1122-JS-DEMO_31". The current file is "app.js". The code in "app.js" includes a button click event listener that fetches data from "api/people.json" and logs the JSON response to the console.
- Middle Window (Browser):** Shows a simple web page with a button labeled "GetData". The browser's developer tools are open, specifically the "Console" tab, which displays the logged JSON data. The data is shown as a promise object that has been fulfilled, containing an array of four objects representing people: {id: 1, name: 'Liang'}, {id: 2, name: '208410331'}, {id: 3, name: 'AAA'}, and {id: 4, name: 'BBB'}.
- Bottom Window (Terminal):** Shows the terminal output with several log entries, including timestamps and command-line interactions related to the project.

=> fetch api vs.xhr

The screenshot displays a comparison between two JavaScript code snippets for fetching JSON data from an API.

Left Side (fetch API):

```
const btn = document.querySelector('.btn');
const url = './api/people.json';

btn.addEventListener('click', () => {
  fetchData(url);
});

function fetchData(url) {
  /*=====promise構成=====*/
  fetch(url)
    /*response.json()抓到response裡面的data*/
    .then((response) => response.json())
    /*回傳son的data*/
    .then((data) => {
      console.log('data', data);
      /*把data送到displayItem function來處理*/
      displayItem(data);
    })
    .catch((err) => {
      console.log(err);
    });
}

function displayItem(data) {
  const displayData = data
    .map((item) => {
      return `
        <p>$<item.name</p>
      `;
    })
    .join('');
  /*產生一個div把資料塞在裡面 */
  const element = document.createElement('div');
  element.innerHTML = displayData;
  document.body.appendChild(element);
}
```

Right Side (XMLHttpRequest):

```
function getData() {
  /* http xhr寫法 */
  const xhr = new XMLHttpRequest();
  console.log('xhr', xhr);
  /*GET取資料,傳到url裡面*/
  xhr.open('GET', url);
  console.log('xhr', xhr);
  xhr.onreadystatechange = function () {
    if (xhr.readyState === 4 && xhr.status === 200) {
      console.log('xhr', xhr);
      /*-----處理資料成字符串過程-----*/
      /*轉成字符串*/
      const data = JSON.parse(xhr.responseText);
      const displayData = data
        .map((item) => {
          return `
            <p>$<item.name</p>
          `;
        })
        .join('');
      /*產生一個div把資料塞在裡面 */
      const element = document.createElement('div');
      element.innerHTML = displayData;
      document.body.appendChild(element);
    } else {
      console.log({
        status: xhr.status,
        text: xhr.statusText,
      });
    }
    xhr.send();
  }
}
```

Output:

A browser window shows the output of the code execution. It displays a table with four rows:

	Liang	208410331	AAA	BBB
1				
2				
3				

The browser's developer tools' console tab shows the value of the `data` variable:

```
data ▶ (4) [{}, {}, {}, {}] ▶
▶ 0: {id: 1, name: 'Liang'}
▶ 1: {id: 2, name: '208410331'}
▶ 2: {id: 3, name: 'AAA'}
▶ 3: {id: 4, name: 'BBB'}
▶ length: 4
▶ [[Prototype]]: Array(0)
```

be3c643 liangyu9103 Thu May 9 20:49:50 2024 +0800 ### W12-P4: Do 3_json_people_fe

W12-P5: Do 3_json_people_async_await

The screenshot shows a developer's environment with multiple windows open:

- Code Editor:** An IDE window showing `app.js` with code related to fetching JSON data and displaying it.
- Browser:** A browser window showing the output of the code execution, including a table with columns "show text", "Liang", "208410331", "AAA", and "BBB".
- Terminal:** A terminal window showing a git log output for the file `app.js`, listing commits from May 9, 2024.
- Git Log:** A detailed view of the git log entries, highlighting specific commits and their details.

Code Editor Content (`app.js`):

```
const btn = document.querySelector('.btn');
const url = './api/people.json';

/*click後執行getData */
btn.addEventListener('click', () => {
    //get url的資料
    fetchData(url);
});

const fetchDataAsync = async (url) => {
    try {
        const response = await fetch(url);
        const data = await response.json();
        console.log('data', data);
        displayItem(data);
    } catch (err) {
        console.log(err);
    }
};

function fetchData(url) {
    /*=====promise架構=====*/
    fetch(url)
        /*response.json()抓到response裡面的data*/
        .then((response) => response.json())
        /*回傳response的資料 */
        .then((data) => {
            console.log('data', data);
            /*把data送到displayItem function來處理*/
            displayItem(data);
        })
        .catch((err) => {
            console.log(err);
        });
}

function displayItem(data) {
}

function getData() {
}
```

Browser Output:

show text	Liang	208410331	AAA	BBB
	Liang	208410331	AAA	BBB

Terminal Git Log:

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-05-08"
```

Git Log Details:

Commit Hash	Author	Date	Message
7c317d6	liangyu9103	Thu May 9 21:09:46 2024 +0800	### W12-P5: Do 3_json_people_as
be3c643	liangyu9103	Thu May 9 20:49:50 2024 +0800	### W12-P4: Do 3_json_people_fe
4269bb0	liangyu9103	Thu May 9 20:19:54 2024 +0800	### W12-P3: Do 3_json_people_xh
53cc848	liangyu9103	Thu May 9 19:46:00 2024 +0800	### W12-P2: Do 2_add_btn
493e73e	liangyu9103	Thu May 9 19:17:21 2024 +0800	### W12-P1: Do 1_Sample-text

W12-P6: w12 all code and log

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-05-08"
```

Commit Hash	Author	Date	Message
7c317d6	liangyu9103	Thu May 9 21:09:46 2024 +0800	### W12-P5: Do 3_json_people_as
be3c643	liangyu9103	Thu May 9 20:49:50 2024 +0800	### W12-P4: Do 3_json_people_fe
4269bb0	liangyu9103	Thu May 9 20:19:54 2024 +0800	### W12-P3: Do 3_json_people_xh
53cc848	liangyu9103	Thu May 9 19:46:00 2024 +0800	### W12-P2: Do 2_add_btn
493e73e	liangyu9103	Thu May 9 19:17:21 2024 +0800	### W12-P1: Do 1_Sample-text