

My Github URL

git config --global user.email "212410368@o365.tku.edu.tw"

git config --global user.name "212410368"

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

W12-P1: Do 1_simple-text

The screenshot shows a Windows desktop environment. On the left, a code editor (VS Code) displays a file named `app.js` with the following content:

```
const xhr = new XMLHttpRequest();
console.log('xhr', xhr);
xhr.onreadystatechange = function() {
  if (xhr.readyState === 4 && xhr.status === 200) {
    const text = document.createElement('p');
    text.textContent = xhr.responseText;
    document.body.appendChild(text);
  } else {
    console.log({
      status: xhr.status,
      text: xhr.statusText,
    });
  }
}
xhr.send();
console.log('hello');
```

A red box highlights the line `if (xhr.readyState === 4 && xhr.status === 200)`. In the bottom right corner of the code editor, there is a message: "Server is Started at port: 5500" and "Live Server (延伸模組)".

In the center, a web browser window titled "AJAX -- liuchengting 212410368" shows the result of the AJAX request. The page content is: "liuchengting.212410368,lorem ipsum dolor, sit amet consectetur adipisicing elit. Corporis totam optio excepturi at vel culpa soluta, quod vitae accusamus necessitatibus error fuga quae dolorum rem unde nihil illo. Ex, ipsa." A red box highlights the first part of the response: "liuchengting.212410368,".

To the right of the browser, the Windows Task Manager is open, showing the process "node.exe" with the command line "C:\Users\liuchengting\Documents\GitHub\W12-P1\index.html". The Task Manager also lists other processes like "System", "File Explorer", and "Google Chrome".

82a7bab 212410368

Thu May 9 19:32:13 2024 +0800

W12-P1: Do 1_simple-text

W12-P2: Do 2_add_btn

=> Network, Header

The screenshot shows a browser window with two tabs: 'index.html' and 'app.js'. The 'index.html' tab displays a simple form with a button labeled 'show text'. The 'app.js' tab shows the JavaScript code for handling this button click. In the browser's developer tools, the Network tab is selected, showing a list of requests. One request, 'sample.txt', is highlighted with a red box. Its details panel is open, showing the Headers and Response sections. The 'Headers' section includes fields like Request URL, Request Method (GET), Status Code (304 Not Modified), and Remote Address (127.0.0.1:5500). The 'Response' section shows the content 'liuchengting.212410368,Lorem ipsum dolor, sit amet consectetur adipisicing elit. Corporis totam optio excepturi at vel culpa soluta, quod vitae accusamus necessitatibus error fuga quae dolorum rem unde nihil illo. Ex ipsa.'

=> Network, Response

This screenshot is similar to the previous one but focuses specifically on the Headers section of the Network tab for the 'sample.txt' request. The 'Headers' panel is expanded, showing detailed information about the request. The 'Request URL' is 'http://127.0.0.1:5500/1122-js-demo-212410368/demo/w12_ajax_68/1_simple-text/api/sample.txt'. The 'Request Method' is 'GET', and the 'Status Code' is '304 Not Modified'. Other headers listed include 'Accept-Ranges: bytes', 'Access-Control-Allow-Origin: *', 'Cache-Control: public, max-age=0', and 'Connection: keep-alive'. The 'Response' section and other parts of the developer tools interface are also visible.

c9b35a8 212410368

Thu May 9 20:00:17 2024 +0800 W12-P2: Do 2_add_btn

W12-P3: Do 3_json_people_xhr

The screenshot shows a development setup with two main windows. On the left is a code editor (VS Code) displaying a file named `app.js` which contains JavaScript code for handling an 'add' button click to fetch JSON data from a local API. On the right is a web browser window showing the results of the AJAX call. The browser's developer tools Network tab is open, highlighting a request to `127.0.0.1:5500/1122-js-demo-212410368/demo/w12_ajax_68/3_json_people_xhr/index.html`. The response tab shows a JSON array of four people objects: `[{"id": 1, "name": "劉政霆"}, {"id": 2, "name": "212410368"}, {"id": 3, "name": "john"}, {"id": 4, "name": "mary"}]`

```
const btn = document.querySelector('.btn');
const url = './api/people.json';
btn.addEventListener('click', () => {
  getData(url);
});

function getData(url) {
  const xhr = new XMLHttpRequest();
  console.log('xhr', xhr);
  xhr.open('GET', url);
  xhr.onreadystatechange = function () {
    console.log('xhr', xhr);
    if (xhr.readyState === 4 && xhr.status === 200) {
      const data = JSON.parse(xhr.responseText);
      const displayData = data.map((item) => {
        return (
          <p>$<item.name</p>
        );
      }).join('');
      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');
```

AJAX -- 劉政
霆, 212410368

Name	Headers	Preview	Response	Initiator	Timing
people.json			[{"id": 1, "name": "劉政霆"}, {"id": 2, "name": "212410368"}, {"id": 3, "name": "john"}, {"id": 4, "name": "mary"}]		

劉政霆
212410368
john
mary
劉政霆
212410368
john
mary

1 requests 295 B transferred

Console

Failed to load resource: the server responded with a status of 404 (Not Found) :5500/favicon.ico:1

xhr0 > XMLHttpRequest
xhr > XMLHttpRequest
xhr > XMLHttpRequest
▶ Object
xhr > XMLHttpRequest
▶ Object

63966f0 212410368

Sun May 12 22:30:45 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 the code editor and browser output for the first example. The code editor displays the file `w12.js-demo-212410368/demo/w12_ajax_68/4_json_people_fetch_api/app.js` with the following code:

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

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

function fetchData(url) {
  fetch(url)
    .then((response) => response.json())
    .then((data) => {
      console.log('data', data);
      displayItems(data);
    })
    .catch((err) => console.log('err', err));
}

function displayItems(data) {
  const displayData = data
    .map((item) => {
      return `<p>${item.name}</p>`;
    })
    .join('');
  const element = document.createElement('div');
  element.innerHTML = displayData;
  document.body.appendChild(element);
}
```

The browser window shows the output of the code. The title bar says "AJAX -- 劉政靈, 212410368". The DevTools Console tab shows the following data:

```
data ▶ (4) [ { } , { } , { } , { } ]
  ▷ 0: { id: 1, name: '劉政靈' }
  ▷ 1: { id: 2, name: '212410368' }
  ▷ 2: { id: 3, name: 'john' }
  ▷ 3: { id: 4, name: 'mary' }
  length: 4
  ▷ [[Prototype]]: Array(0)
```

The browser status bar indicates "啟用 Windows" (Enable Windows).

=> fetch api vs.xhr

The screenshot shows the code editor and browser output for the second example. The code editor displays the file `w12.js-demo-212410368/demo/w12_ajax_68/4_json_people_fetch_api/app.js` with the following code:

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

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

function fetchData(url) {
  fetch(url)
    .then((response) => response.json())
    .then((data) => {
      console.log('data', data);
      displayItems(data);
    })
    .catch((err) => console.log('err', err));
}

function displayItems(data) {
  const displayData = data
    .map((item) => {
      return `<p>${item.name}</p>`;
    })
    .join('');
  const element = document.createElement('div');
  element.innerHTML = displayData;
  document.body.appendChild(element);
}

function getData(url) {
  const xhr = new XMLHttpRequest();
  console.log('xhr', xhr);
  xhr.open('GET', url);
  console.log('xhr', xhr);
  xhr.onreadystatechange = function () {
    if (xhr.readyState === 4 && xhr.status === 200) {
      const data = JSON.parse(xhr.responseText); // ...
      const displayData = data
        .map((item) => {
          return `<p>${item.name}</p>`;
        })
        .join('');
      const element = document.createElement('div');
      element.innerHTML = displayData;
      document.body.appendChild(element);
    }
  }
}
```

The browser window shows the output of the code. The title bar says "AJAX -- 劉政靈, 212410368". The DevTools Console tab shows the following data:

```
data ▶ (4) [ { } , { } , { } , { } ]
  ▷ 0: { id: 1, name: '劉政靈' }
  ▷ 1: { id: 2, name: '212410368' }
  ▷ 2: { id: 3, name: 'john' }
  ▷ 3: { id: 4, name: 'mary' }
  length: 4
  ▷ [[Prototype]]: Array(0)
```

The browser status bar indicates "啟用 Windows" (Enable Windows).

cfe637f 212410368

Sun May 12 22:31:32 2024 +0800 ### W12-P4: Do 3_json_people_fetch_api

W12-P5: Do 3_json_people_async_await

The screenshot shows a code editor and a browser window. The code editor on the left contains JavaScript code for handling an 'async/await' button click. The browser window on the right displays the resulting page with the title 'AJAX -- 劉政霆, 212410368'. It shows three items: 廖政霆, 212410368, john, and mary. The browser's developer tools console shows the fetched JSON data: [{id: 1, name: '劉政霆'}, {id: 2, name: '212410368'}, {id: 3, name: 'john'}, {id: 4, name: 'mary'}]. A red box highlights the fetch data logic in the code editor, and another red box highlights the JSON data in the browser's console.

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

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

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

function displayItems(data) {
  const displayData = data
    .map(item => {
      return `
        <p>${item.name}</p>
      `;
    })
    .join('');
  const element = document.createElement('div');
  element.innerHTML = displayData;
  document.body.appendChild(element);
}

// function getData(url) {
//   const xhr = new XMLHttpRequest();
//   console.log('xhr', xhr);
// }
```

AJAX -- 劉政霆, 212410368

劉政霆
212410368
john
mary

```
Live reload enabled.
data ▾ (4) [ {<...>, <...>, <...>} ]
  ▷ 0: {id: 1, name: '劉政霆'}
  ▷ 1: {id: 2, name: '212410368'}
  ▷ 2: {id: 3, name: 'john'}
  ▷ 3: {id: 4, name: 'mary'}
  length: 4
  [[Prototype]]: Array(0)
```

6b00e51 212410368

Sun May 12 22:32:05 2024 +0800 ### W12-P5: Do 3_json_people_async_await

W12-P6: git logs for W12 ..

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

562f477 212410368	Sun May 12 22:33:38 2024 +0800	W12-P6: git logs for W12
6b00e51 212410368	Sun May 12 22:32:05 2024 +0800	### W12-P5: Do 3_json_people_async_await
cfe637f 212410368	Sun May 12 22:31:32 2024 +0800	### W12-P4: Do 3_json_people_fetch_api
63966f0 212410368	Sun May 12 22:30:45 2024 +0800	W12-P3: Do 3_json_people_xhr
c9b35a8 212410368	Thu May 9 20:00:17 2024 +0800	W12-P2: Do 2_add_btn
82a7bab 212410368	Thu May 9 19:32:13 2024 +0800	W12-P1: Do 1_simple-text