

## My Github URL

## W12-P1: Do 1\_simple-text

The screenshot shows a development environment with VS Code on the left and a web browser on the right. In VS Code, the file `index.html` is open, showing an AJAX request using `XMLHttpRequest`. The `onreadystatechange` event listener is highlighted with a red box. The browser shows the page content: "AJAX -- htchung, 123456789" followed by Lorem ipsum text. The DevTools Console shows the response of the AJAX request, with the `response` and `status` properties highlighted by red boxes.

```
$ git log --pretty=format:"%h%x09an%x09ad%x09s" --after="2024-06-12"
90bd9eb Sky001 Thu Jun 13 09:59:15 2024 +0800 W12-P1: Do 1_simple-text
```

## W12-P2: Do 2\_add\_btn

## =&gt; Network, Header

The screenshot shows a development environment with VS Code on the left and a web browser on the right. In VS Code, the file `index.html` is open, showing an AJAX request using `XMLHttpRequest`. The `onreadystatechange` event listener is highlighted with a red box. The browser shows the page content: "AJAX -- htchung, 123456789" followed by Lorem ipsum text. A button labeled "show text" is visible. The DevTools Network tab shows the response of the AJAX request, with the `response` property highlighted by a red box.

## =&gt; Network, Response

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-12"
8aa4daa Sky001 Thu Jun 13 10:01:31 2024 +0800 W12-P2: Do 2_add_btn
```

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-12"
fcb128b Sky001 Thu Jun 13 10:10:22 2024 +0800 W12-P3: Do 3_json_people_xhr
```

## W12-P4: Do 3\_json\_people\_fetch\_api

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

The left side shows a VS Code editor with a JavaScript file `app.js` containing the following code:

```

1 const btn = document.querySelector('.btn');
2 const url = './api/people.json';
3
4 btn.addEventListener('click', () => {
5   fetchData(url);
6 });
7
8 function fetchData(url) {
9   fetch(url)
10    .then(response => console.log('response json', response.json()))
11    .catch(err => console.log(err));
12 }
13
14 > function getData(url) { ...
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

```

The right side shows a web browser at `127.0.0.1:5502/demo/w12_ajax_xx/4_json_people_fetch...` with the title "AJAX -- htchung, 123456789". A "show text" button is visible. The browser's DevTools console shows the following log:

```

response json
  Promise {<pending>}
    [[Prototype]]: Promise
    [[PromiseState]]: "fulfilled"
    [[PromiseResult]]: Array(4)
      0: {id: 1, name: 'htchung'}
      1: {id: 2, name: '123456789'}
      2: {id: 3, name: 'john'}
      3: {id: 4, name: 'mary'}
      length: 4
      [[Prototype]]: Array(0)

```

=> `fetch api vs.xhr`

The left side shows a VS Code editor with two JavaScript files side-by-side. The left file `app.js` uses `fetch` and `displayItems` to show data. The right file `app.js` uses `XMLHttpRequest` and `getData` to show data.

The right side shows a web browser at `127.0.0.1:5502...` with the title "AJAX -- htchung, 123456789". A "show text" button is visible. The browser's DevTools console shows the following log:

```

data
  (4) [{"id": 1, "name": "htchung"}, {"id": 2, "name": "123456789"}, {"id": 3, "name": "john"}, {"id": 4, "name": "mary"}]
    length: 4
    [[Prototype]]: Array(0)

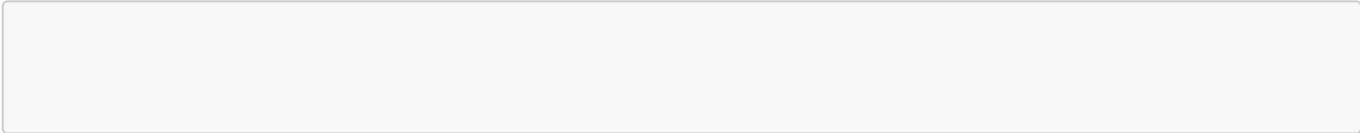
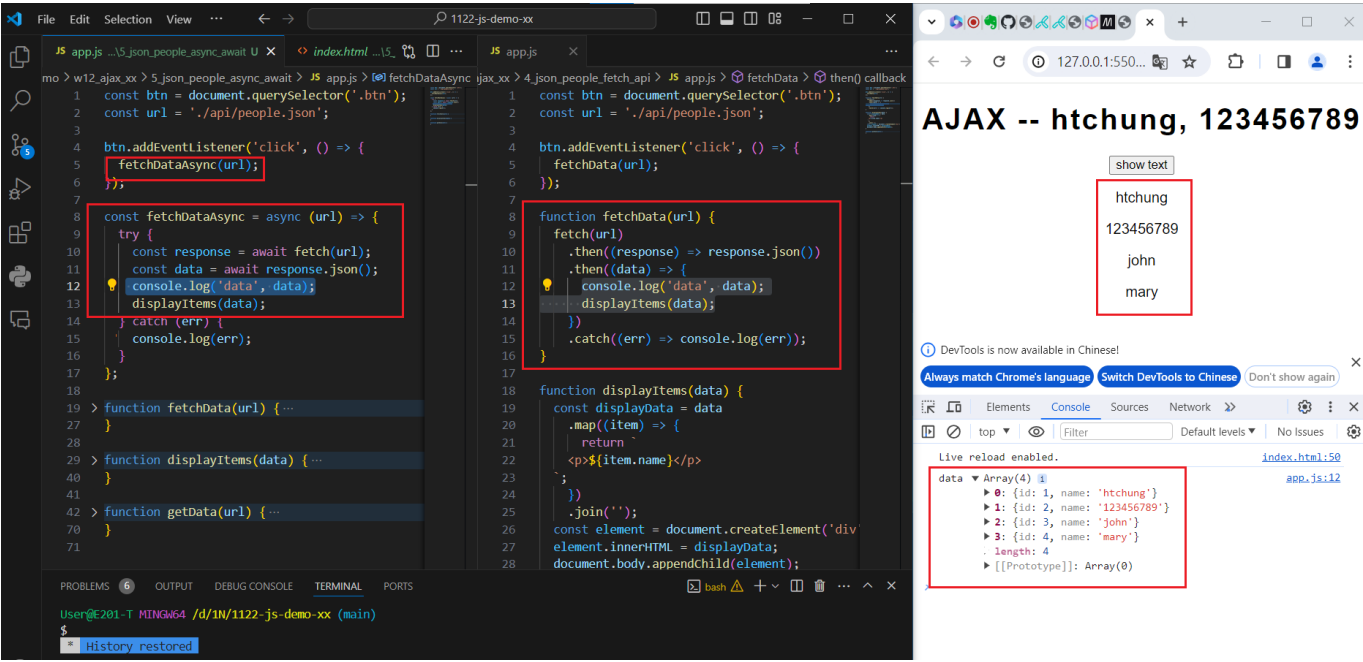
```

```

$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-12"
65a31f2 Sky001 Thu Jun 13 10:15:24 2024 +0800 W12-P4: Do 3_json_people_fetch_api

```

## W12-P5: Do 3\_json\_people\_async\_await



W11-P5: git logs for W11

```
git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2024-06-12"
65a31f2 Sky001 Thu Jun 13 10:15:24 2024 +0800 W12-P4: Do 3_json_people_fetch_api
fcb128b Sky001 Thu Jun 13 10:10:22 2024 +0800 W12-P3: Do 3_json_people_xhr
8aa4daa Sky001 Thu Jun 13 10:01:31 2024 +0800 W12-P2: Do 2_add_btn
90bd9eb Sky001 Thu Jun 13 09:59:15 2024 +0800 W12-P1: Do 1_simple-text
```