CWEB280 -wk11 – LO7 – Client/Server Communication

# Fetch – Xml Http Request

## Fetch API <https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API>

If you ever heard of AJAX or used jQuery.ajax(), youwill be familiar with Xml Http Requests. In newr browsers the older browser specific way of making XmlHttpRequest an new api call Fetch is more powerful and flexible with the following benefits

* provides a generic definition of Request and Response objects
* allow Request and Response object to be used wherever they are needed in the future
* Fetch is available in pretty much any context you might need it

Fetch options

* method: GET (default), POST, PUT, DELETE, etc.
* mode: no-cors, cors (default), same-origin
* cache: default, no-cache, reload, force-cache, only-if-cached
* credentials: include (allows api to set cookies), same-origin (default), omit
* redirect: manual, follow (default), error
* referrerPolicy: no-referrer, no-referrer-when-downgrade (default), origin, origin-when-cross-origin, same-origin, strict-origin, strict-origin-when-cross-origin(newer default), unsafe-url
* body: json string data sent to server – no available for GET method
* headers: custom headers sent to the server  
   – for JSON body you need the header 'Content-Type': 'application/json'

## Using Fetch <https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch>

Very Simple Example

fetch('http://example.com/movies.json')

.then(response => response.json())

.then(data => console.log(data));

More Complex Post Example

// Example POST method implementation:

async function postData(url = '', data = {}) {

// Default options are marked with \*

const response = await fetch(url, {

method: 'POST', // \*GET, POST, PUT, DELETE, etc.

mode: 'cors', // no-cors, \*cors, same-origin

cache: 'no-cache', // \*default, no-cache, reload, force-cache, only-if-cached

credentials: 'same-origin', // include, \*same-origin, omit

headers: {

'Content-Type': 'application/json'

// 'Content-Type': 'application/x-www-form-urlencoded',

},

redirect: 'follow', // manual, \*follow, error

referrerPolicy: 'no-referrer', // no-referrer, \*no-referrer-when-downgrade, origin, origin-when-cross-origin, same-origin, strict-origin, strict-origin-when-cross-origin, unsafe-url

body: JSON.stringify(data) // body data type must match "Content-Type" header

});

return response.json(); // parses JSON response into native JavaScript objects

}

postData('https://example.com/answer', { answer: 42 })

.then(data => {

console.log(data); // JSON data parsed by `data.json()` call

});

Vue CLI: <https://cli.vuejs.org/guide/>

Json server: <https://developer.school/tutorials/how-to-use-vue-js-json-server-and-axios>

Vue cli project structure: <https://itnext.io/how-to-structure-my-vue-js-project-e4468db005ac>

Vue prop: <https://vuejs.org/v2/guide/components-props.html>