## W10 - FETCH APIs ##

Fetch API provides JavaScript Interface for accessing and manipulating parts of HTTP pipeline (request/responses)

Global method: fetch()

- a better alternative that can be used by technologies like service workers.
- a single logical place to define HTTP-related concepts like CORS (cross-origin resource sharing)

→ Promise resolves with ok property set to false if response isn't in 200-299 range

Basic Fetch() request:

```
fetch('http://example.com/movies.json')
  .then(resonse => response.json())
  .then(data => console.log(data));
```

A init Object as the second parameter

Example POST method (also in week10.js)

```
async function postData(url='', data={}) {
    // default options marked with astrisk
    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-cac
        credentials: 'same-origin', // include, *same-origin, omit
        headers: {
          'Content-type': 'application/json'
        },
        redirect: 'follow', // manual, *follow, error
        referrerPolicy: 'no-referrer', //
        body: JSON.stringify(data) // body type must match content header
    });
    return response.JSON(); // parses JSON response
}
```

cont on next page

## W10 - Fetch APIs ##

```
postData('https://example.com/answer', { answer: 42 })
  .then (data => {
     console.log(data); // JSON data parsed
  });
```

no-cors allow limited set of headers in request

- Accept • Accept-language • Content-Language
- Content-Type: application/x-www-form-urlencoded
  multipart/form-data
  text/plain

Sending Request w/ credentials included

```
fetch ('https://example.com'), {
    credentials: 'include' // 'same-origin', 'omit'
  });
```

Uploading JSON data

```
fetch ('https://example.com/profile', {
    method: 'POST', // or 'PUT'
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify (data),
  })
  .then (response => response.json())
  .then (data => {  console.log ('Success:', data);
  .catch ((error) => {  console.log ('Error:', error);});
```

UPLOADING A FILE → using `<input type="file" multiple />`

Processing a text file line by line