

## Module : (10) [Web API's (i)]

### ① Fetch () / Axios

- Used to make HTTP requests to get or send data.

// FETCH

// GET request using Fetch

```
Fetch ("https://jsonplaceholder.typicode.com/posts/1")
```

- then (response  $\Rightarrow$  response.json())
- then (data  $\Rightarrow$  console.log (data))
- catch (error  $\Rightarrow$  console.error ("Error:", error))

// AXIOS

```
axios.get ("https://jsonplaceholder.typicode.com/posts/1")
```

- then (response  $\Rightarrow$  console.log (response.data))
- catch (error  $\Rightarrow$  console.error ("Error:", error))

### ② localStorage / sessionStorage

- Store data in the browser.

// localStorage

```
localStorage.setItem ("username", "ng24"); // store  
console.log (localStorage.getItem ("username")); // retrieve  
localStorage.removeItem ("username"); // remove
```

// Session Storage

```
sessionStorage.setItem ("token", "123abc");  
console.log (sessionStorage.getItem ("token"));  
sessionStorage.clear();
```

// removes all session storage item



### ③ navigator.geolocation

- Get's user's current location (latitude & longitude).

```

if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition (
    (position) => {
    console.log("Latitude:", position.coords.
      latitude);
    console.log("Longitude:", position.coords.
      longitude);
    },
    (error) => {
    console.error("Error:", error);
    });
} else {
  console.log("Geolocation not supported by
    browser");
}

```

### ④ Intersection Observer

- Detects when an element enters the viewport  
(useful for lazy-loading).

```

const box = document.querySelector(".box");
const observer = new IntersectionObserver((entries)
=> {
  entries.forEach((entry) => {
    if (entry.isIntersecting) {
      console.log("Box is visible!");
      observer.observe(box);
    }
  });
});

```



### ⑤ AbortController

Cancel a fetch request if it's taking too long.

```
const controller = new AbortController();
```

```
const signal = controller.signal
```

```
fetch ("https://jsonplaceholder.typicode.com/posts",  
      {signal})
```

```
  .then (res => res.json())
```

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

```
  .catch (err => {
```

```
    if (err.name === "AbortError") console.log ("Fetch  
    aborted!");
```

```
    else console.error(err); });
```

// cancel the request after 2 seconds

```
setTimeout ((s => controller.abort(), 2000);
```

### ⑥ requestAnimationFrame()

Create smooth animations in the browser.

```
const box = document.querySelector (".box");
```

```
let pos = 0;
```

```
function animate() {
```

```
  pos += 2;
```

```
  box.style.left = pos + "px";
```

```
  if (pos < 300) {
```

```
    requestAnimationFrame (animate);
```

```
  }  
  animate();
```