

Fetch API

Laiba Imran

What is Fetch API?

- fetch is a modern JavaScript method for making HTTP requests to web servers.
- Replaces older methods like XMLHttpRequest.
- Used for getting data from APIs, sending data to servers, or updating resources.

Why Fetch API?

Some main issues with XMLHttpRequest:

1. Complex syntax
2. Lack of Promise Support
3. Cumbersome Error Handling

```
const xhr = new XMLHttpRequest();
xhr.open('GET', 'https://api.example.com/data', true);

xhr.onreadystatechange = function() {
  if (xhr.readyState === 4) {
    if (xhr.status === 200) {
      const data = JSON.parse(xhr.responseText);
      console.log('Data received:', data);
    } else {
      switch (xhr.status) {
        case 400:
          console.error('Bad Request: The server could not understand the request.');
```

break;

```
        case 404:
          console.error('Not Found: The requested resource could not be found.');
```

break;

```
        .....
        default:
          console.error(`Error: Received status code ${xhr.status}`);
      }
    }
  }
};
xhr.send();
```

```
fetch('https://api.example.com/data')
  .then(response => {
    if (!response.ok)
    {
      throw new Error(`HTTP error! status: ${response.status}`);
    }
    return response.json();
  })
  .then(data => console.log('Data received:', data))
  .catch(error => console.error('Fetch error:', error));
```

Basic Syntax of Fetch

```
fetch(url, options)  
  .then(response => response.json())  
  .then(data => console.log(data))  
  .catch(error => console.error('Error:', error));
```

Handling Responses

- **Response Methods:**

- `.json()`: Parse JSON data.
- `.text()`: Retrieve response as plain text.

- **Example:**

```
fetch('https://api.example.com/data')  
  .then(response => response.text())  
  .then(data => console.log(data))  
  .catch(...);
```

Fetch with GET Request

```
fetch('https://api.example.com/data')  
  .then(response => response.json())  
  .then(data => console.log(data))  
  .catch(..);
```


Fetch with POST Request

```
fetch('https://api.example.com/data', {  
  method: 'POST',  
  headers: { 'Content-Type': 'application/json' },  
  body: JSON.stringify({ key: 'value' })  
})  
  .then(response => response.json())  
  .then(data => console.log(data));
```

JSON.stringify converts the JavaScript object { key: 'value' } into a JSON-formatted string: "{ \"key\": \"value\" }".

Async/Await with Fetch

```
async function fetchData() {  
  const url = 'https://api.example.com/data';  
  try {  
    const response = await fetch(url);  
    const data = await response.json();  
    console.log(data);  
  }  
  catch (error) {  
    console.error('Error:', error);  
  }  
}  
fetchData();
```

Resources

Go over https://www.w3schools.com/js/js_api_fetch.asp