JavaScript Async Fetch

Back to JS page

Table of Contents

- JavaScript Async Fetch
  - Fetching a File
    - Example 1
  - The Response Object
    - Example 2
  - Fetching JSON
    - Example 3
  - Loading Multiple Files
    - Example 4
  - Handling Errors
    - Example 5
  - Document
  - Reference

Fetching a File

The `fetch()` method loads a file from a web server asynchronously. It returns a Promise that resolves to a Response object:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Async Fetch</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Fetching a File</h4>
<p id="demo"></p>

<script>
async function getData() {
  const response = await fetch('https://jsonplaceholder.typicode.com/posts/1');
  const data = await response.json();
  document.getElementById("demo").innerHTML =
    "Title: " + data.title + "<br>" +
    "Body: " + data.body;
}

getData();
</script>

</body>
</html>
```

Example 1

Result View Example

The Response Object

The Response object has useful properties like `ok`, `status`, and `url`:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Async Fetch</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>The Response Object</h4>
<p id="demo"></p>

<script>
async function getResponse() {
  const response = await fetch('https://jsonplaceholder.typicode.com/users/1');

  document.getElementById("demo").innerHTML =
    "ok: " + response.ok + "<br>" +
    "status: " + response.status + "<br>" +
    "statusText: " + response.statusText + "<br>" +
    "url: " + response.url;

  const data = await response.json();
  document.getElementById("demo").innerHTML += "<br>Name: " + data.name;
}

getResponse();
</script>

</body>
</html>
```

Example 2

Result View Example

Fetching JSON

The `response.json()` method parses the response body as JSON:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Async Fetch</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Fetching JSON</h4>
<p id="demo"></p>

<script>
async function fetchJSON() {
  const response = await fetch('https://jsonplaceholder.typicode.com/users');
  const users = await response.json();

  let text = "";
  for (let user of users) {
    text += user.name + " - " + user.email + "<br>";
  }
  document.getElementById("demo").innerHTML = text;
}

fetchJSON();
</script>

</body>
</html>
```

Example 3

Result View Example

Loading Multiple Files

You can load multiple files in parallel using `Promise.all()`:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Async Fetch</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Loading Multiple Files</h4>
<p id="demo"></p>

<script>
async function loadMultiple() {
  const [user, posts] = await Promise.all([
    fetch('https://jsonplaceholder.typicode.com/users/1'),
    fetch('https://jsonplaceholder.typicode.com/posts?userId=1')
  ]);

  const userData = await user.json();
  const postsData = await posts.json();

  document.getElementById("demo").innerHTML =
    "User: " + userData.name + "<br>" +
    "Email: " + userData.email + "<br><br>" +
    "Posts (" + postsData.length + "):<br>";

  for (let post of postsData.slice(0, 3)) {
    document.getElementById("demo").innerHTML += "- " + post.title + "<br>";
  }
}

loadMultiple();
</script>

</body>
</html>
```

Example 4

Result View Example

Handling Errors

Always handle errors when using `fetch()`. Use `try/catch` to catch network errors and check `response.ok` for HTTP errors:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Async Fetch</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Handling Errors</h4>
<p id="demo"></p>

<script>
async function fetchWithErrorHandling() {
  try {
    const response = await fetch('https://jsonplaceholder.typicode.com/users/invalid');

    if (!response.ok) {
      throw new Error("HTTP error! Status: " + response.status);
    }

    const data = await response.json();
    document.getElementById("demo").innerHTML = "Name: " + data.name;
  } catch(error) {
    document.getElementById("demo").innerHTML = "Error caught: " + error.message;
  }
}

fetchWithErrorHandling();
</script>

</body>
</html>
```

Example 5

Result View Example

Document

Document in project

You can Download PDF file.

Reference

- W3Schools JavaScript Async Fetch