Response: error() static method

- Syntax
  - Parameters
  - Return value
- Examples
  - Returning a network error from a service worker
- Specifications
- Browser compatibility
- See also

{{APIRef("Fetch API")}}{{AvailableInWorkers}}

The `error()` static method of the {{domxref("Response")}} interface returns a new `Response` object associated with a network error.

This is mainly useful when writing service workers: it enables a service worker to send a response from a {{domxref("ServiceWorkerGlobalScope.fetch_event", "fetch")}} event handler that will cause the {{domxref("Window/fetch", "fetch()")}} call in the main app code to reject the promise.

An error response has its {{domxref("Response.type","type")}} set to `error`.

Syntax

```
Response.error()
```

Parameters

None.

Return value

A {{domxref("Response")}} object.

Examples

Returning a network error from a service worker

Suppose a web app has a service worker, which contains the following `fetch` event handler:

```
// service-worker.js

self.addEventListener("fetch", (event) => {
  const url = new URL(event.request.url);
  if (url.pathname === "/salamander.jpg") {
    event.respondWith(Response.error());
  }
});
```

With this service worker, all fetch requests from the app will pass through the service worker to the network, except for requests to fetch "salamander.jpg", which will reject. This means that the following main thread code would throw an error, and the `catch` handler will run.

```
// main.js

const image = document.querySelector("#image");

try {
  const response = await fetch("salamander.jpg");
  const blob = await response.blob();
  const objectURL = URL.createObjectURL(blob);
  image.src = objectURL;
} catch (e) {
  console.error(e);
}
```

Specifications

{{Specifications}}

Browser compatibility

{{Compat}}

See also

- ServiceWorker API
- HTTP access control (CORS)
- HTTP

□□□ #2
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ □□□ □□□ 2026-10-31 03:27:53 UTC