Response: type property

- Value
- Examples
  - A basic response
  - A CORS response
  - An opaque response
  - An error response
- Specifications
- Browser compatibility
- See also

The `type` read-only property of the Response interface contains the type of the response. The type determines whether scripts are able to access the response body and headers.

Value

A string, which may be any of the following values:

- `basic`
  - : This applies in any of the following cases:
    - The request is same-origin.
    - The requested URL's scheme is `data:`.
    - The request's mode is `navigate` or `websocket`.

    With this type, all response headers are exposed except Set-Cookie.
- `cors`
  - : The request was cross-origin and was successfully processed using CORS. With this type, only CORS-safelisted response headers are exposed.
- `error`
  - : A network error occurred. The status property is set to `0`, body is `null`, headers are empty and immutable.

This is the type of response returned by Response.error(). A response of this type is not returned by a call to fetch(), because if a network error occurs, the promise is rejected.

- `opaque`
  - : A response to a cross-origin request whose mode was set to `no-cors`. The status property is set to `0`, body is `null`, headers are empty and immutable.
- `opaqueredirect`
  - : A response to a request whose redirect option was set to `manual`, and which was redirected by the server. The status property is set to `0`, body is `null`, headers are empty and immutable.

Examples

A basic response

The following same-origin request will return a `basic` response:

```
const response = await fetch("flowers.jpg");


console.log(response.type); // "basic"
```

A CORS response

Assuming `https://example.org` is not the requester's origin, and that the server responds with the appropriate CORS headers, this request will return a `cors` response:

```
const response = await fetch("https://example.org/flowers.jpg");


console.log(response.type); // "cors"
```

An opaque response

The following request is made with the `no-cors` option, so it returns an `opaque` response:

```
const response = await fetch("https://example.org/flowers.jpg", {
  mode: "no-cors",
});

console.log(response.type); // "opaque"
console.log(response.body); // null
console.log(response.status); // 0
```

An error response

The following code uses Response.error() to create an `error` response:

```
const response = Response.error();

console.log(response.type); // "error"
console.log(response.body); // null
console.log(response.status); // 0
```

Specifications

Browser compatibility

See also

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

□□□ #3
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ cocobook-sync □□□ 2026-11-04 17:34:58 UTC