

# Origin header

**Baseline** Widely available

The HTTP **Origin** request header indicates the **origin** (scheme, hostname, and port) that *caused* the request. For example, if a user agent needs to request resources included in a page, or fetched by scripts that it executes, then the origin of the page may be included in the request.

Header type	Request header
Forbidden request header	Yes

## Syntax

HTTP

```
Origin: null
Origin: <scheme>://<hostname>
Origin: <scheme>://<hostname>:<port>
```

## Directives

**null**

The origin is "privacy sensitive", or is an **opaque origin** (specific cases are listed in the [description](#) section).

**<scheme>**

The protocol that is used. Usually, it is the HTTP protocol or its secured version, HTTPS.

**<hostname>**

The domain name or the IP address of the origin server.

**<port>** Optional

Port number on which the server is listening. If no port is given, the default port for the requested service is implied from the scheme (e.g., **80** for an HTTP URL).

# Description

The `Origin` header is similar to the `Referer` header, but does not disclose the path, and may be `null`. It is used to provide the security context for the origin request, except in cases where the origin information would be sensitive or unnecessary.

Broadly speaking, user agents add the `Origin` request header to:

- `cross origin` requests.
- `same-origin` requests except for `GET` or `HEAD` requests (i.e., they are added to same-origin `POST`, `OPTIONS`, `PUT`, `PATCH`, and `DELETE` requests).

There are some exceptions to the above rules; for example, if a cross-origin `GET` or `HEAD` request is made in `no-cors mode`, the `Origin` header will not be added.

The `Origin` header value may be `null` in a number of cases, including (non-exhaustively):

- Origins whose `scheme` is not one of `http`, `https`, `ftp`, `ws`, `wss`, or `gopher` (including `blob`, `file` and `data`).
- Cross-origin images and media data, including that in `<img>`, `<video>` and `<audio>` elements.
- Documents created programmatically using `createDocument()`, generated from a `data:` URL, or that do not have a creator browsing context.
- Redirects across origins.
- Documents served with the `Content-Security-Policy` `sandbox` directive whose value doesn't include `allow-same-origin`.
- `iframes` with a `sandbox` attribute whose value doesn't include `allow-same-origin`.
- Responses that are network errors.
- `Referrer-Policy` set to `no-referrer` for non-`cors` request modes (e.g., basic form posts).

**Note:** There is a more detailed listing of cases that may return `null` on Stack Overflow: [When do browsers send the Origin header? When do browsers set the origin to null?](#) ↗

## Examples

```
HTTP
```

```
Origin: https://developer.mozilla.org
```

```
HTTP
```

```
Origin: https://developer.mozilla.org:80
```

## Specifications

**Specification**

The Web Origin Concept

# section-7 [↗](#)

Fetch

# origin-header [↗](#)

## Browser compatibility

## See also

- [Host](#)
- [Referer](#)
- [Same-origin policy](#)
- [When do browsers send the Origin header? When do browsers set the origin to null? \[↗\]\(#\) \(Stack Overflow\)](#)



Your blueprint for a better internet.