X-Frame-Options

The **`X-Frame-Options`** HTTP response header can be used to indicate whether or not a browser should be allowed to render a page in a `<frame>`, `<iframe>`, `<embed>` or `<object>`. Sites can use this to avoid click-jacking attacks, by ensuring that their content is not embedded into other sites.

The added security is provided only if the user accessing the document is using a browser that supports `X-Frame-Options`.

> **Note:** The `Content-Security-Policy` HTTP header has a `frame-ancestors` directive which obsoletes this header for supporting browsers.

Header type	Response header
Forbidden header name	no

Syntax

There are two possible directives for `X-Frame-Options`:

```
X-Frame-Options: DENY
X-Frame-Options: SAMEORIGIN
```

Directives

If you specify `DENY`, not only will the browser attempt to load the page in a frame fail when loaded from other sites, attempts to do so will fail when loaded from the same site. On the other hand, if you specify `SAMEORIGIN`, you can still use the page in a frame as long as the site including it in a frame is the same as the one serving the page.

`DENY`

> The page cannot be displayed in a frame, regardless of the site attempting to do so.

`SAMEORIGIN`

> The page can only be displayed in a frame on the same origin as the page itself. The spec leaves it up to browser vendors to decide whether this option applies to the top level, the parent, or the whole chain, although it is argued that the option is not very useful unless all ancestors are also in the same origin (see bug 725490). Also see Browser compatibility for support details.

`ALLOW-FROM uri`

> This is an obsolete directive that no longer works in modern browsers. Don't use it. In supporting legacy browsers, a page can be displayed in a frame only on the specified origin *uri*. Note that in the legacy Firefox implementation this still suffered from the same problem as `SAMEORIGIN` did — it doesn't check the frame ancestors to see if they are in the same origin. The `Content-Security-Policy` HTTP header has a `frame-ancestors` directive which you can use instead.

Examples

> **Note:** Setting X-Frame-Options inside the `<meta>` element is useless! For instance, `<meta http-equiv="X-Frame-Options" content="deny">` has no effect. Do not use it! `X-Frame-Options` works only by setting through the HTTP header, as in the examples below.

Configuring Apache

To configure Apache to send the `X-Frame-Options` header for all pages, add this to your site's configuration:

```
Header always set X-Frame-Options "SAMEORIGIN"
```

To configure Apache to set the `X-Frame-Options` DENY, add this to your site's configuration:

```
Header set X-Frame-Options "DENY"
```

Configuring Nginx

To configure Nginx to send the `X-Frame-Options` header, add this either to your http, server or location configuration:

```
add_header X-Frame-Options SAMEORIGIN always;
```

Configuring IIS

To configure IIS to send the `X-Frame-Options` header, add this to your site's `Web.config` file:

```
<system.webServer>
  ...

  <httpProtocol>
    <customHeaders>
      <add name="X-Frame-Options" value="SAMEORIGIN" />
    </customHeaders>
  </httpProtocol>

  ...
</system.webServer>
```

Or see this Microsoft support article on setting this configuration using the IIS Manager user interface.

Configuring HAProxy

To configure HAProxy to send the `X-Frame-Options` header, add this to your front-end, listen, or backend configuration:

```
rspadd X-Frame-Options:\ SAMEORIGIN
```

Alternatively, in newer versions:

```
http-response set-header X-Frame-Options SAMEORIGIN
```

Configuring Express

To configure Express to send the `X-Frame-Options` header, you can use helmet which uses frameguard to set the header. Add this to your server configuration:

```
const helmet = require('helmet');
const app = express();
app.use(helmet.frameguard({ action: 'SAMEORIGIN' }));
```

Alternatively, you can use frameguard directly:

```
const frameguard = require('frameguard')
app.use(frameguard({ action: 'SAMEORIGIN' }))
```

Specifications

Specification

Specification

HTTP Header Field X-Frame-Options
section-2

Browser compatibility

Report problems with this compatibility data on GitHub

	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	WebView Android	Chrome Android	Firefox for Android
`X-Frame-Options`	Chrome 4	Edge 12	Firefox 4	Internet Explorer 8	Opera 10.5	Safari 4	WebView Android Yes	Chrome Android Yes	Firefox for Androi
ALLOW-FROM	Chrome No	Edge 12–79	Firefox 18–70	Internet Explorer 8	Opera No	Safari No	WebView Android No	Chrome Android No	Firefox Andro
SAMEORIGIN	Chrome Yes	Edge 12	Firefox Yes	Internet Explorer 8	Opera Yes	Safari Yes	WebView Android Yes	Chrome Android Yes	Firefox for Androic

Full support No support Compatibility unknown See implementation notes.

See also

- `Content-Security-Policy` directive `frame-ancestors`
- HTTP Header Field X-Frame-Options - RFC 7034
- ClickJacking Defenses - IEBlog
- Combating ClickJacking with X-Frame-Options - IEInternals

Last modified: Apr 14, 2022, by MDN contributors