
mdn web docs



Ready, set, build on Contentful. Use our platform to tackle builds and scale with both speed and confidence.

Mozilla ads

Don't want to see ads?

History

The `History` interface allows manipulation of the browser session history, that is the pages visited in the tab or frame that the current page is loaded in.

There is only one instance of `history` (It is a singleton.) accessible via the global object `history`.

Note: This interface is only available on the main thread (`Window`). It cannot be accessed in `Worker` or `Worklet` contexts.

Instance properties

The `History` interface doesn't inherit any property.

`length` Read only

Returns an `Integer` representing the number of elements in the session history, including the currently loaded page. For example, for a page loaded in a new tab this property returns `1`.

`scrollRestoration`

Allows web applications to explicitly set default scroll restoration behavior on history navigation. This property can be either `auto` or `manual`.

`state` Read only

Returns an `any` value representing the state at the top of the history stack. This is a way to look at the state without having to wait for a `popstate` event.

Instance methods

The `History` interface doesn't inherit any methods.

`back()`

This asynchronous method goes to the previous page in session history, the same action as when the user clicks the browser's `Back` button. Equivalent to `history.go(-1)`.

Calling this method to go back beyond the first page in the session history has no effect and doesn't raise an exception.

`forward()`

This asynchronous method goes to the next page in session history, the same action as when the user clicks the browser's `Forward` button; this is equivalent to `history.go(1)`.

Calling this method to go forward beyond the most recent page in the session history has no effect and doesn't raise an exception.

`go()`

Asynchronously loads a page from the session history, identified by its relative location to the current page, for example `-1` for the previous page or `1` for the next page. If you specify an out-of-bounds value (for instance, specifying `-1` when there are no previously-visited pages in the session history), this method silently has no effect. Calling `go()` without parameters or a value of `0` reloads the current page.

`pushState()`

Pushes the given data onto the session history stack with the specified title (and, if provided, URL). The data is treated as opaque by the DOM; you may specify any JavaScript object that can be serialized. Note that all browsers but Safari currently ignore the title parameter. For more information, see [Working with the History API](#).

`replaceState()`

Updates the most recent entry on the history stack to have the specified data, title, and, if provided, URL. The data is treated as opaque by the DOM; you may specify any JavaScript object that can be serialized. Note that all browsers but Safari currently ignore the title parameter. For more information, see [Working with the History API](#).

Specifications

Specification
HTML Standard
the-history-interface

Browser compatibility

[Report problems with this compatibility data on GitHub](#)

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	
<code>History</code>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android	✓ Opera 10.1 Android	✓ Safari on 1 iOS	✓ Samsung 1.0 Internet	✓ 1
<code>back</code>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 12.1	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android	✓ Opera 12.1 Android	✓ Safari on 1 iOS	✓ Samsung 1.0 Internet	✓ 1

	Desktop					Mobile					
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	
forward	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 12.1	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android	✓ Opera 12.1 Android	✓ Safari on 1 iOS	✓ Samsung 1.0 Internet	✓ 1
go	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 12.1	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android	✓ Opera 12.1 Android	✓ Safari on 1 iOS	✓ Samsung 1.0 Internet	✓ 1
length	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 12.1	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android	✓ Opera 12.1 Android	✓ Safari on 1 iOS	✓ Samsung 1.0 Internet	✓ 1
pushState	✓ Chrome 5	✓ Edge 12	✓ Firefox 4	✓ Opera 11.5	✓ Safari 5	✓ Chrome 18 Android	✓ Firefox for 4 Android	✓ Opera 11.5 Android	✓ Safari on 4 iOS	✓ Samsung 1.0 Internet	✓ 1
Whether the <code>unused</code> parameter is used	✗ Chrome No	✗ Edge No	✗ Firefox No	✗ Opera No	✓ Safari 5	✗ Chrome No Android	✗ Firefox No for Android	✗ Opera No Android	✓ Safari 4 on iOS	✗ Samsung No Internet	✗ 1
replaceState	✓ Chrome 5	✓ Edge 12	✓ Firefox 4	✓ Opera 11.5	✓ Safari 5	✓ Chrome 18 Android	✓ Firefox for 4 Android	✓ Opera 11.5 Android	✓ Safari on 4 iOS	✓ Samsung 1.0 Internet	✓ 1
Whether the <code>unused</code> parameter is used	✗ Chrome No	✗ Edge No	✗ Firefox No	✗ Opera No	✓ Safari 5	✗ Chrome No Android	✗ Firefox No for Android	✗ Opera No Android	✓ Safari 4 on iOS	✗ Samsung No Internet	✗ 1
scrollRestoration	✓ Chrome 46	✓ Edge 79	✓ Firefox 46	✓ Opera 33	✓ Safari 11	✓ Chrome 46 Android	✓ Firefox for 46 Android	✓ Opera 33 Android	✓ Safari 11 on iOS	✓ Samsung 5.0 Internet	✓ 1
state	✓ Chrome 19	✓ Edge 12	✓ Firefox 4	✓ Opera 12.1	✓ Safari 6	✓ Chrome 25 Android	✓ Firefox for 4 Android	✓ Opera 12.1 Android	✓ Safari on 6 iOS	✓ Samsung 1.5 Internet	✓ 1

Tip: you can click/tap on a cell for more information.

✓ Full support ✗ No support ⚠ Non-standard. Check cross-browser support before using. 🚫 Deprecated. Not for use in new websites. ⓘ See implementation notes.

See also

- [history](#) global object

This page was last modified on Mar 29, 2023 by [MDN contributors](#).