FH JOANNEUM
University of Applied Sciences

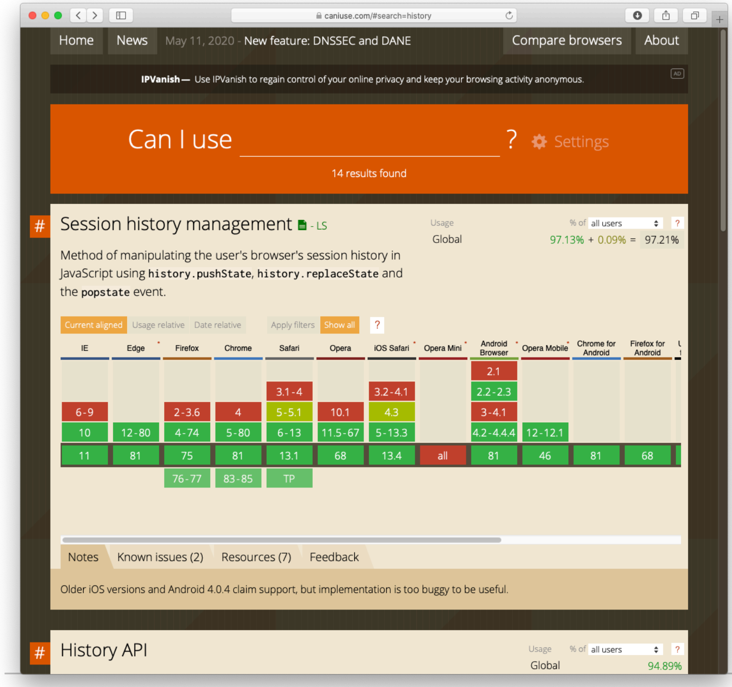
Single Page Applications (SPA)

No page reload, modify DOM

John Feiner

APPLIED COMPUTER SCIENCES

What's Available?


caniuse.com/#search=history

Home | News | May 11, 2020 - New feature: DNSSEC and DANE | Compare browsers | About

IPVanish — Use IPVanish to regain control of your online privacy and keep your browsing activity anonymous.

Can I use ______ ? Settings

14 results found

Session history management - LS

Usage Global — % of all users: 97.13% + 0.09% = 97.21%

Method of manipulating the user's browser's session history in JavaScript using `history.pushState`, `history.replaceState` and the `popstate` event.

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

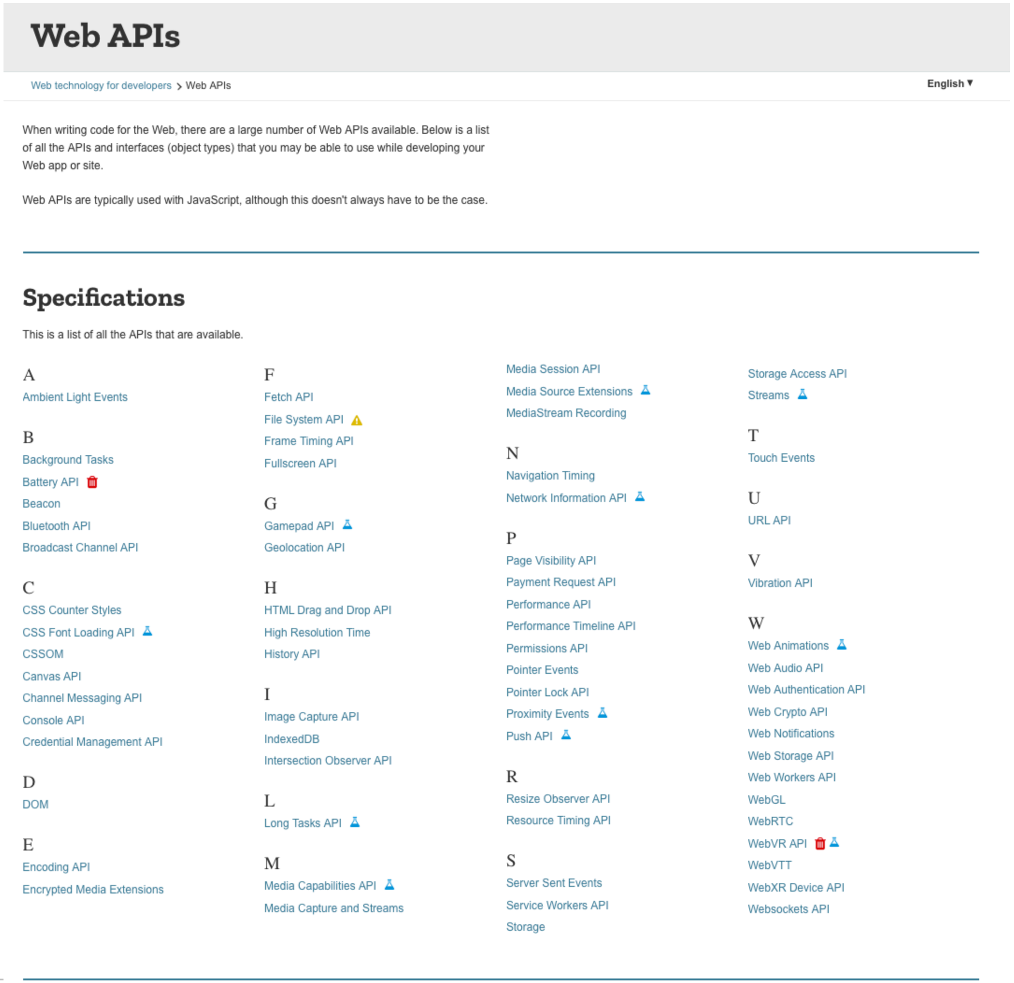
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android
								2.1			
				3.1-4		3.2-4.1		2.2-2.3			
6-9		2-3.6	4	5-5.1	10.1	4.3		3-4.1			
10	12-80	4-74	5-80	6-13	11.5-67	5-13.3		4.2-4.4.4	12-12.1		
11	81	75	81	13.1	68	13.4	all	81	46	81	68
		76-77	83-85	TP							

Notes | Known issues (2) | Resources (7) | Feedback

Older iOS versions and Android 4.0.4 claim support, but implementation is too buggy to be useful.

History API

Usage Global — % of all users: 94.89%


Web APIs

Web technology for developers > Web APIs — English ▾

When writing code for the Web, there are a large number of Web APIs available. Below is a list of all the APIs and interfaces (object types) that you may be able to use while developing your Web app or site.

Web APIs are typically used with JavaScript, although this doesn't always have to be the case.

Specifications

This is a list of all the APIs that are available.

A
- Ambient Light Events

B
- Background Tasks
- Battery API
- Beacon
- Bluetooth API
- Broadcast Channel API

C
- CSS Counter Styles
- CSS Font Loading API
- CSSOM
- Canvas API
- Channel Messaging API
- Console API
- Credential Management API

D
- DOM

E
- Encoding API
- Encrypted Media Extensions

F
- Fetch API
- File System API
- Frame Timing API
- Fullscreen API

G
- Gamepad API
- Geolocation API

H
- HTML Drag and Drop API
- High Resolution Time
- History API

I
- Image Capture API
- IndexedDB
- Intersection Observer API

L
- Long Tasks API

M
- Media Capabilities API
- Media Capture and Streams
- Media Session API
- Media Source Extensions
- MediaStream Recording

N
- Navigation Timing
- Network Information API

P
- Page Visibility API
- Payment Request API
- Performance API
- Performance Timeline API
- Permissions API
- Pointer Events
- Pointer Lock API
- Proximity Events
- Push API

R
- Resize Observer API
- Resource Timing API

S
- Server Sent Events
- Service Workers API
- Storage
- Storage Access API
- Streams

T
- Touch Events

U
- URL API

V
- Vibration API

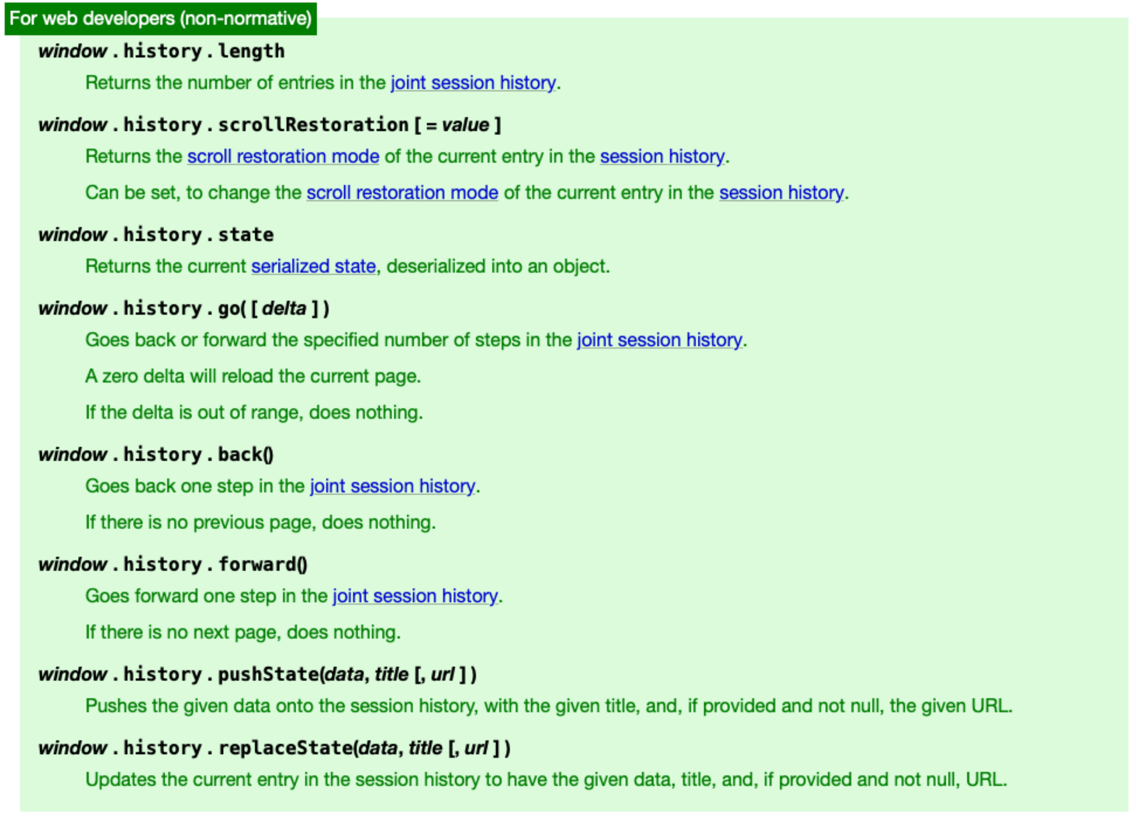
W
- Web Animations
- Web Audio API
- Web Authentication API
- Web Crypto API
- Web Notifications
- Web Storage API
- Web Workers API
- WebGL
- WebRTC
- WebVR API
- WebVTT
- WebXR Device API
- Websockets API

https://caniuse.com

https://developer.mozilla.org/en-US/docs/Web/API

History API

Modify your session history e.g. the URL



For web developers (non-normative)

window . history . length
Returns the number of entries in the joint session history.

window . history . scrollRestoration [= *value*]
Returns the scroll restoration mode of the current entry in the session history.
Can be set, to change the scroll restoration mode of the current entry in the session history.

window . history . state
Returns the current serialized state, deserialized into an object.

window . history . go([*delta*])
Goes back or forward the specified number of steps in the joint session history.
A zero delta will reload the current page.
If the delta is out of range, does nothing.

window . history . back()
Goes back one step in the joint session history.
If there is no previous page, does nothing.

window . history . forward()
Goes forward one step in the joint session history.
If there is no next page, does nothing.

window . history . pushState(*data*, *title* [, *url*])
Pushes the given data onto the session history, with the given title, and, if provided and not null, the given URL.

window . history . replaceState(*data*, *title* [, *url*])
Updates the current entry in the session history to have the given data, title, and, if provided and not null, URL.

https://developer.mozilla.org/en-US/docs/Web/API/History_API