

[Extensions APIs](#)**Extension APIs**[Apps APIs](#)

Chrome APIs

Chrome provides extensions with many special-purpose APIs like `chrome.runtime` and `chrome.alarms`.

Stable APIs

Name	Description	Since
accessibilityFeatures	Use the <code>chrome.accessibilityFeatures</code> API to manage Chrome's accessibility features. This API relies on the ChromeSetting prototype of the type API for getting and setting individual accessibility features. In order to get feature states the extension must request <code>accessibilityFeatures.read</code> permission. For modifying feature state, the extension needs <code>accessibilityFeatures.modify</code> permission. Note that <code>accessibilityFeatures.modify</code> does not imply <code>accessibilityFeatures.read</code> permission.	37
alarms	Use the <code>chrome.alarms</code> API to schedule code to run periodically or at a specified time in the future.	27
bookmarks	Use the <code>chrome.bookmarks</code> API to create, organize, and otherwise manipulate bookmarks. Also see Override Pages , which you can use to create a custom Bookmark Manager page.	27
browserAction	Use browser actions to put icons in the main Google Chrome toolbar, to the right of the address bar. In addition to its icon , a browser action can have a tooltip , a badge , and a popup .	27
browsingData	Use the <code>chrome.browsingData</code> API to remove	27

	browsing data from a user's local profile.	
certificateProvider	Use this API to expose certificates to the platform which can use these certificates for TLS authentications.	46
commands	Use the commands API to add keyboard shortcuts that trigger actions in your extension, for example, an action to open the browser action or send a command to the extension.	27
contentSettings	Use the <code>chrome.contentSettings</code> API to change settings that control whether websites can use features such as cookies, JavaScript, and plugins. More generally speaking, content settings allow you to customize Chrome's behavior on a per-site basis instead of globally.	27
contextMenus	Use the <code>chrome.contextMenus</code> API to add items to Google Chrome's context menu. You can choose what types of objects your context menu additions apply to, such as images, hyperlinks, and pages.	27
cookies	Use the <code>chrome.cookies</code> API to query and modify cookies, and to be notified when they change.	27
debugger	The <code>chrome.debugger</code> API serves as an alternate transport for Chrome's remote debugging protocol . Use <code>chrome.debugger</code> to attach to one or more tabs to instrument network interaction, debug JavaScript, mutate the DOM and CSS, etc. Use the Debuggee <code>tabId</code> to target tabs with <code>sendCommand</code> and route events by <code>tabId</code> from <code>onEvent</code> callbacks.	27
declarativeContent	Use the <code>chrome.declarativeContent</code> API to take actions depending on the content of a page, without requiring permission to read the page's content.	33
desktopCapture	Desktop Capture API that can be used to capture content of screen, individual windows or tabs.	34
devtools.inspectedWindow	Use the <code>chrome.devtools.inspectedWindow</code> API to interact with the inspected window: obtain the tab ID for the inspected page, evaluate the code in the context of the inspected window, reload the page, or obtain the list of resources within the page.	27

devtools.network	Use the <code>chrome.devtools.network</code> API to retrieve the information about network requests displayed by the Developer Tools in the Network panel.	27
devtools.panels	Use the <code>chrome.devtools.panels</code> API to integrate your extension into Developer Tools window UI: create your own panels, access existing panels, and add sidebars.	27
documentScan	Use the <code>chrome.documentScan</code> API to discover and retrieve images from attached paper document scanners.	44
downloads	Use the <code>chrome.downloads</code> API to programmatically initiate, monitor, manipulate, and search for downloads.	31
enterprise.deviceAttributes	Use the <code>chrome.enterprise.deviceAttributes</code> API to read device attributes. Note: This API is only available to extensions force-installed by enterprise policy.	46
enterprise.platformKeys	Use the <code>chrome.enterprise.platformKeys</code> API to generate hardware-backed keys and to install certificates for these keys. The certificates will be managed by the platform and can be used for TLS authentication, network access or by other extension through <code>chrome.platformKeys</code> .	37
events	The <code>chrome.events</code> namespace contains common types used by APIs dispatching events to notify you when something interesting happens.	27
extension	The <code>chrome.extension</code> API has utilities that can be used by any extension page. It includes support for exchanging messages between an extension and its content scripts or between extensions, as described in detail in Message Passing .	27
extensionTypes	The <code>chrome.extensionTypes</code> API contains type declarations for Chrome extensions.	39
fileBrowserHandler	Use the <code>chrome.fileBrowserHandler</code> API to extend the Chrome OS file browser. For example, you can use this API to enable users to upload files to your website.	27
fileSystemProvider	Use the <code>chrome.fileSystemProvider</code> API to	40

	create file systems, that can be accessible from the file manager on Chrome OS.	
fontSettings	Use the <code>chrome.fontSettings</code> API to manage Chrome's font settings.	27
gcm	Use <code>chrome.gcm</code> to enable apps and extensions to send and receive messages through the Google Cloud Messaging Service .	35
history	Use the <code>chrome.history</code> API to interact with the browser's record of visited pages. You can add, remove, and query for URLs in the browser's history. To override the history page with your own version, see Override Pages .	27
i18n	Use the <code>chrome.i18n</code> infrastructure to implement internationalization across your whole app or extension.	27
identity	Use the <code>chrome.identity</code> API to get OAuth2 access tokens.	29
idle	Use the <code>chrome.idle</code> API to detect when the machine's idle state changes.	27
input.ime	Use the <code>chrome.input.ime</code> API to implement a custom IME for Chrome OS. This allows your extension to handle keystrokes, set the composition, and manage the candidate window.	27
instanceID	Use <code>chrome.instanceID</code> to access the Instance ID service.	46
management	The <code>chrome.management</code> API provides ways to manage the list of extensions/apps that are installed and running. It is particularly useful for extensions that override the built-in New Tab page.	27
networking.config	Use the <code>networking.config</code> API to authenticate to captive portals.	43
notifications	Use the <code>chrome.notifications</code> API to create rich notifications using templates and show these notifications to users in the system tray.	28
omnibox	The omnibox API allows you to register a keyword with Google Chrome's address bar, which is also known as the omnibox.	27

pageAction	Use the <code>chrome.pageAction</code> API to put icons in the main Google Chrome toolbar, to the right of the address bar. Page actions represent actions that can be taken on the current page, but that aren't applicable to all pages. Page actions appear grayed out when inactive.	27
pageCapture	Use the <code>chrome.pageCapture</code> API to save a tab as MHTML.	27
permissions	Use the <code>chrome.permissions</code> API to request declared optional permissions at run time rather than install time, so users understand why the permissions are needed and grant only those that are necessary.	27
platformKeys	Use the <code>chrome.platformKeys</code> API to access client certificates managed by the platform. If the user or policy grants the permission, an extension can use such a certificate in its custom authentication protocol. E.g. this allows usage of platform managed certificates in third party VPNs (see <code>chrome.vpnProvider</code>).	45
power	Use the <code>chrome.power</code> API to override the system's power management features.	27
printerProvider	The <code>chrome.printerProvider</code> API exposes events used by print manager to query printers controlled by extensions, to query their capabilities and to submit print jobs to these printers.	44
privacy	Use the <code>chrome.privacy</code> API to control usage of the features in Chrome that can affect a user's privacy. This API relies on the ChromeSetting prototype of the type API for getting and setting Chrome's configuration.	27
proxy	Use the <code>chrome.proxy</code> API to manage Chrome's proxy settings. This API relies on the ChromeSetting prototype of the type API for getting and setting the proxy configuration.	27
runtime	Use the <code>chrome.runtime</code> API to retrieve the background page, return details about the manifest, and listen for and respond to events in the app or extension lifecycle. You can also use this API to convert the relative path of URLs to fully-qualified URLs.	27

sessions	Use the <code>chrome.sessions</code> API to query and restore tabs and windows from a browsing session.	37
storage	Use the <code>chrome.storage</code> API to store, retrieve, and track changes to user data.	27
system.cpu	Use the <code>system.cpu</code> API to query CPU metadata.	32
system.memory	The <code>chrome.system.memory</code> API.	32
system.storage	Use the <code>chrome.system.storage</code> API to query storage device information and be notified when a removable storage device is attached and detached.	30
tabCapture	Use the <code>chrome.tabCapture</code> API to interact with tab media streams.	31
tabs	Use the <code>chrome.tabs</code> API to interact with the browser's tab system. You can use this API to create, modify, and rearrange tabs in the browser.	27
topSites	Use the <code>chrome.topSites</code> API to access the top sites that are displayed on the new tab page.	27
tts	Use the <code>chrome.tts</code> API to play synthesized text-to-speech (TTS). See also the related ttsEngine API, which allows an extension to implement a speech engine.	27
ttsEngine	Use the <code>chrome.ttsEngine</code> API to implement a text-to-speech(TTS) engine using an extension. If your extension registers using this API, it will receive events containing an utterance to be spoken and other parameters when any extension or Chrome App uses the tts API to generate speech. Your extension can then use any available web technology to synthesize and output the speech, and send events back to the calling function to report the status.	27
types	The <code>chrome.types</code> API contains type declarations for Chrome.	27
vpnProvider	Use the <code>chrome.vpnProvider</code> API to implement a VPN client.	43
wallpaper	Use the <code>chrome.wallpaper</code> API to change the ChromeOS wallpaper.	31

webNavigation	Use the <code>chrome.webNavigation</code> API to receive notifications about the status of navigation requests in-flight.	27
webRequest	Use the <code>chrome.webRequest</code> API to observe and analyze traffic and to intercept, block, or modify requests in-flight.	27
windows	Use the <code>chrome.windows</code> API to interact with browser windows. You can use this API to create, modify, and rearrange windows in the browser.	27

Beta APIs

These APIs are only available in the Chrome Beta and Dev channels:

Name	Description
declarativeNetRequest	The <code>chrome.declarativeNetRequest</code> API is used to block or redirect network requests by specifying declarative rules.
declarativeWebRequest	Note: <i>this API is currently on hold, without concrete plans to move to stable.</i> Use the <code>chrome.declarativeWebRequest</code> API to intercept, block, or modify requests in-flight. It is significantly faster than the <code>chrome.webRequest</code> API because you can register rules that are evaluated in the browser rather than the JavaScript engine, which reduces roundtrip latencies and allows higher efficiency.

Dev APIs

These APIs are only available in the Chrome Dev channel:

Name	Description
automation	The <code>chrome.automation</code> API allows developers to access the automation (accessibility) tree for the browser. The tree resembles the DOM tree, but only exposes the <i>semantic</i> structure of a page. It can be used to programmatically interact with a page by examining names, roles, and states, listening for events, and performing actions on nodes.
processes	Use the <code>chrome.processes</code> API to interact with the browser's processes.
signedInDevices	Use the <code>chrome.signedInDevices</code> API to get a list of devices signed into chrome with the same account as the current profile.

Experimental APIs

Chrome also has **experimental APIs**, some of which will become supported APIs in future releases of Chrome.

API conventions

Unless the doc says otherwise, methods in the chrome.* APIs are **asynchronous**: they return immediately, without waiting for the operation to finish. If you need to know the outcome of an operation, then you pass a callback function into the method. For more information, watch this video:

Google Chrome Extensions: Extension API Design



Content available under the [CC-BY 3.0 license](#)

Google Terms of Service Privacy Policy Report
a content bug

Add us on 