



Extend the Browser

What are Extensions?

Get Started Tutorial

Overview

Developer Guide

Samples

Help

FAQ

What's New?

Google Groups

Stack Overflow

Web Store Publishing and Distribution

Mobile Chrome

Frequently Asked Questions

If you don't find an answer to your question here, try the [Chrome Web Store FAQ](#), the [\[google-chrome-extension\] tag on Stack Overflow](#), the [chromium-extensions group](#), or the [store help](#).

General

- [What are Google Chrome Extensions?](#)
- [How can I set up Chrome for extension development?](#)
- [What technologies are used to write extensions for Chrome?](#)
- [Are extensions fetched from the web every time the browser is loaded?](#)
- [How do I determine which version of Chrome is deployed to which channel?](#)

Capabilities

- [Can extensions make cross-domain Ajax requests?](#)
- [Can extensions use 3rd party web services?](#)
- [Can extensions encode/decode JSON data?](#)
- [Can extensions store data locally?](#)
- [Can extensions use OAuth?](#)
- [Can extensions create UI outside of the rendered web page?](#)

- [Can extensions listen to clicks on Chrome tabs and navigation buttons?](#)
- [Can two extensions communicate with each other?](#)
- [Can extensions use Google Analytics?](#)
- [Can extensions modify chrome:// URLs?](#)
- [Can extensions open browser/page action popups without user interaction?](#)
- [Can extensions keep popups open after the user clicks away from them?](#)
- [Can extensions be notified when they are installed/uninstalled?](#)

Development

- [How do I build a UI for my extension?](#)
- [How much data can I store in localStorage?](#)
- [How do I create an options menu for my application?](#)
- [What debugging tools are available to extension developers?](#)
- [Why do wildcard matches not work for top level domains \(TLDs\)?](#)
- [Why does the management API not fire events when my extension is installed/uninstalled?](#)
- [How can an extension determine whether it is running for the first time?](#)

Features and bugs

- [I think I've found a bug! How do I make sure it gets fixed?](#)
- [I have a feature request! How can I report it?](#)


General

What are Google Chrome Extensions?

Google Chrome Extensions are applications that run inside the Chrome browser and provide additional functionality, integration with third party websites or services, and customized browsing experiences.

How can I set up Chrome for extension development?

As long as you are using a version of Chrome that supports extensions, you already have everything you need to start writing an extension of your own. You can start by turning on Developer mode.

Click the Chrome menu icon  and select **Extensions** from the **Tools** menu. Ensure that the "Developer mode" checkbox in the top right-hand corner is checked. Now you can reload extensions, load an unpacked directory of files as if it were a packaged extension, and more. For a complete tutorial, see [Getting Started](#).

What technologies are used to write extensions for Chrome?

Extensions are written using the same standard web technologies that developers use to create websites. HTML is used as a content markup language, CSS is used for styling, and JavaScript for scripting. Because Chrome supports HTML5 and CSS3, developers can use the latest open web technologies such as canvas and CSS animations in their extensions. Extensions also have access to several [JavaScript APIs](#) that help perform functions like JSON encoding and interacting with the browser.

Are extensions fetched from the web every time the browser is loaded?

Extensions are downloaded by the Chrome browser upon install, and are subsequently run off of the local disk in order to speed up performance. However, if a new version of the extension is pushed online, it will be automatically downloaded in the background to any users who have the extension installed. Extensions may also make requests for remote content at any time, in order to interact with a web service or pull new content from the web.

How do I determine which version of Chrome is deployed to which channel?

To determine which version of Chrome is currently available on each of the different platforms, visit omahaproxy.appspot.com. On that site you will see data in a format similar to:

```
cf,dev,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
cf,beta,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
cf,stable,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
linux,dev,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
linux,beta,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
linux,stable,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
mac,dev,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
mac,beta,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
mac,stable,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
win,canary,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
win,dev,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
win,beta,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
win,stable,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
cros,dev,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
cros,beta,###.###.###,mm/dd/yy,mm/dd/yy,###,###,###
```

Each line represents information about a different platform and channel combination. The listed platforms are **cf** (Google Chrome Frame), **linux**, **mac**, **win**, and **cros** (Google Chrome OS). The listed channels are **canary**, **dev**, **beta**, and **stable**. The two four-part numbers after the channel represent the current and previous versions of Chrome deployed to that platform-channel combination. The rest of the information is metadata about when the releases were first pushed, as well as revision numbers associated with each build.

Capabilities

Can extensions make cross-domain Ajax requests?

Yes. Extensions can make cross-domain requests. See [this page](#) for more information.

Can extensions use 3rd party web services?

Yes. Extensions are capable of making cross-domain Ajax requests, so they can call remote APIs directly. APIs that provide data in JSON format are particularly easy to use.

Can extensions encode/decode JSON data?

Yes, because V8 (Chrome's JavaScript engine) supports `JSON.stringify` and `JSON.parse` natively, you may use these functions in your extensions [as described here](#) without including any additional JSON libraries in your code.

Can extensions store data locally?

Yes, extensions can use [localStorage](#) to store string data permanently. Using Chrome's built-in JSON functions, you can store complex data structures in `localStorage`. For extensions that need to execute SQL queries on their stored data, Chrome implements [client side SQL databases](#), which may be used as well.

Can extensions use OAuth?

Yes, there are extensions that use OAuth to access remote data APIs. Most developers find it convenient to use a [JavaScript OAuth library](#) in order to simplify the process of signing OAuth requests.

Can extensions create UI outside of the rendered web page?

Yes, your extension may add buttons to the Chrome browser's user interface. See [browser actions](#) and [page actions](#) for more information.

An extension may also create popup notifications, which exist outside of the browser window. See the [desktop notifications](#) documentation for more details.

Can extensions listen to clicks on Chrome tabs and navigation buttons?

No. Extensions are limited to listening to the events described in the [API documentation](#).

Can two extensions communicate with each other?

Yes, extensions may pass messages to other extensions. See the [message passing documentation](#) for more information.

Can extensions use Google Analytics?

Yes, since extensions are built just like websites, they can use [Google Analytics](#) to track usage. However, you must modify the tracking code to pull an HTTPS version of the Google Analytics library. See [this tutorial](#) for more information on doing this.

Can extensions modify `chrome://` URLs?

No. The extensions APIs have been designed to minimize backwards compatibility issues that can arise when new versions of the browser are pushed. Allowing content scripts on `chrome://` URLs would mean that developers would begin to rely on the DOM, CSS, and JavaScript of these pages to stay the same. In the best case, these pages could not be updated as quickly as they are being updated right now. In the

worst case, it could mean that an update to one of these pages could cause an extension to break, causing key parts of the browser to stop working for users of that extension.

The reason that **replacing the content** hosted at these URLs entirely is allowed is because it forces an extension developer to implement all of the functionality they want without depending on the browser's internal implementation to stay the same.

Can extensions open browser/page action popups without user interaction?

No, popups can only be opened if the user clicks on the corresponding page or browser action. An extension cannot open its popup programatically.

Can extensions keep popups open after the user clicks away from them?

No, popups automatically close when the user focuses on some portion of the browser outside of the popup. There is no way to keep the popup open after the user has clicked away.

Can extensions be notified when they are installed/uninstalled?

You can listen to the **runtime.onInstalled** event to be notified when your extension is installed or updated, or when Chrome itself is updated. There is no corresponding event for when your extension is uninstalled.

Development

How do I build a UI for my extension?

Extensions use HTML and CSS to define their user interfaces, so you can use standard form controls to build your UI, or style the interface with CSS, as you would a web page. Additionally, extensions can add **some limited UI elements to Chrome itself**.

How much data can I store in localStorage?

Extensions can store up to 5MB of data in localStorage.

How do I create an options menu for my application?

You can let users set options for your extension by creating an **options page**, which is a simple HTML page that will be loaded when a user clicks the "options" button for your extension. This page can read and write settings to localStorage, or even send options to a web server so that they can be persisted across browsers.

What debugging tools are available to extension developers?

Chrome's built-in developer tools can be used to debug extensions as well as web pages. See this **tutorial on debugging extensions** for more information.

Why do wildcard matches not work for top level domains (TLDs)?

You cannot use wildcard match patterns like `http://google.*/*` to match TLDs (like `http://google.es` and `http://google.fr`) due to the complexity of actually restricting such a match to only the desired domains.

For the example of `http://google.*/*`, the Google domains would be matched, but so would `http://google.someotherdomain.com`. Additionally, many sites do not own all of the TLDs for their domain. For an example, assume you want to use `http://example.*/*` to match `http://example.com` and `http://example.es`, but `http://example.net` is a hostile site. If your extension has a bug, the hostile site could potentially attack your extension in order to get access to your extension's increased privileges.

You should explicitly enumerate the TLDs that you wish to run your extension on.

Why does the management API not fire events when my extension is installed/uninstalled?

The **management API** was intended to help create new tab page replacement extensions. It was not intended to fire install/uninstall events for the current extension.

How can an extension determine whether it is running for the first time?

You can listen to the **runtime.onInstalled** event. See [this FAQ entry](#).

Features and bugs

I think I've found a bug! How do I make sure it gets fixed?

While developing an extension, you may find behavior that does not match the extensions documentation and may be the result of a bug in Chrome. The best thing to do is to make sure an appropriate issue report is filed, and the Chromium team has enough information to reproduce the behavior.

The steps you should follow to ensure this are:

1. Come up with a *minimal* test extension that demonstrates the issue you wish to report. This extension should have as little code as possible to demonstrate the bug—generally this should be 100 lines of code or less. Many times, developers find that they cannot reproduce their issues this way, which is a good indicator that the bug is in their own code.
2. Search the issue tracker at <http://crbug.com> to see whether someone has reported a similar issue. Most issues related to extensions are filed under **component=Platform>Extensions**, so to look for an extension bug related to the `chrome.tabs.executeScript` function (for example), search for `"component=Platform>Extensions Type=Bug chrome.tabs.executeScript"`, which will give you [this list of results](#).
3. If you find a bug that describes your issue, click the star icon to be notified when the bug receives an update. *Do not respond to the bug to say "me too" or ask "when will this be fixed?"*; such updates can cause hundreds of emails to be sent. Add a comment only if you have information (such as a better test case or a suggested fix) that is likely to be helpful.

4. If you found no appropriate bug to star, file a new issue report at <http://crbug.com/new>. Be as explicit as possible when filling out this form: choose a descriptive title, explain the steps to reproduce the bug, and describe the expected and actual behavior. Attach your test example to the report and add screenshots if appropriate. The easier your report makes it for others to reproduce your issue, the greater chance that your bug will be fixed promptly.
5. Wait for the bug to be updated. Most new bugs are triaged within a week, although it can sometimes take longer for an update. *Do not reply to the bug to ask when the issue will be fixed.* If your bug has not been modified after two weeks, please post a message to the [discussion group](#) with a link back to your bug.
6. If you originally reported your bug on the discussion group and were directed to this FAQ entry, reply to your original thread with a link to the bug you starred or reported. This will make it easier for others experiencing the same issue to find the correct bug.

I have a feature request! How can I report it?

If you identify a feature (especially if it's related to an experimental API) that could be added to improve the extension development experience, make sure an appropriate request is filed in the issue tracker.

The steps you should follow to ensure this are:

1. Search the issue tracker at <http://crbug.com> to see whether someone has requested a similar feature. Most requests related to extensions are filed under **component=Platform>Extensions**, so to look for an extension feature request related to keyboard shortcuts (for example), search for "**component=Platform>Extensions Type=Feature shortcuts**", which will give you [this list of results](#).
2. If you find a ticket that matches your request, click the star icon to be notified when the bug receives an update. *Do not respond to the bug to say "me too" or ask "when will this be implemented?"*; such updates can cause hundreds of emails to be sent.
3. If you found no appropriate ticket to star, file a new request at <http://crbug.com/new>. Be as detailed as possible when filling out this form: choose a descriptive title and explain exactly what feature you would like and how you plan to use it.
4. Wait for the ticket to be updated. Most new requests are triaged within a week, although it can sometimes take longer for an update. *Do not reply to the ticket to ask when the feature will be added.* If your ticket has not been modified after two weeks, please post a message to the [discussion group](#) with a link back to your request.
5. If you originally reported your request on the discussion group and were directed to this FAQ entry, reply to your original thread with a link to the ticket you starred or opened. This will make it easier for others with the same request to find the correct ticket.

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

[Google](#) [Terms of Service](#) [Privacy Policy](#) [Report](#)
[a content bug](#)

Add us on 