The [**OpenSearch description format**](http://www.opensearch.org/Specifications/OpenSearch/1.1#OpenSearch_description_document) lets a website describe a search engine for itself, so that a browser or other client application can use that search engine. OpenSearch is supported by (at least) Firefox, Edge, Internet Explorer, Safari, and Chrome. (See [Reference Material](https://developer.mozilla.org/en-US/docs/Web/OpenSearch#Reference_Material) for links to other browsers' documentation.)

Firefox also supports additional features not in the OpenSearch standard, such as search suggestions and the <SearchForm> element. This article focuses on creating OpenSearch-compatible search plugins that support these additional Firefox features.

OpenSearch description files can be advertised as described in [Autodiscovery of search plugins](https://developer.mozilla.org/en-US/docs/Web/OpenSearch#Autodiscovery_of_search_plugins), and can be installed programmatically as described in [Adding search engines from web pages](https://developer.mozilla.org/en/Adding_search_engines_from_web_pages).

**OpenSearch description file**[**Section**](https://developer.mozilla.org/en-US/docs/Web/OpenSearch#OpenSearch_description_file)

The XML file that describes a search engine follows the basic template below. Sections in *[square brackets]* should be customized for the specific plugin you're writing.

<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/"

xmlns:moz="http://www.mozilla.org/2006/browser/search/">

<ShortName>[SNK]</ShortName>

<Description>[Search engine full name and summary]</Description>

<InputEncoding>[UTF-8]</InputEncoding>

<Image width="16" height="16" type="image/x-icon">[https://example.com/favicon.ico]</Image>

<Url type="text/html" template="[searchURL]">

<Param name="[key name]" value="{searchTerms}"/>

<!-- other Params if you need them… -->

<Param name="[other key name]" value="[parameter value]"/>

</Url>

<Url type="application/x-suggestions+json" template="[suggestionURL]"/>

<moz:SearchForm>[https://example.com/search]</moz:SearchForm>

</OpenSearchDescription>

**ShortName**

A short name for the search engine. It must be **16 or fewer character**s of plain text, with no HTML or other markup.

**Description**

A brief description of the search engine. It must be **1024 or fewer characters** of plain text, with no HTML or other markup.

**InputEncoding**

The [character encoding](https://developer.mozilla.org/en-US/docs/Glossary/character_encoding) to use when submitting input to the search engine.

**Image**

URI of an icon for the search engine. When possible, include a 16×16 image of type image/x-icon (such as /favicon.ico) and a 64×64 image of type image/jpeg or image/png.

The URI may also use the [data: URI scheme](https://developer.mozilla.org/en-US/docs/Web/HTTP/data_URIs). (You can generate a data: URI from an icon file at [The data: URI kitchen](http://software.hixie.ch/utilities/cgi/data/data).)

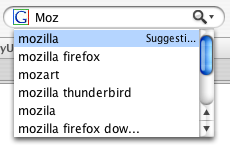
<Image height="16" width="16" type="image/x-icon">https://example.com/favicon.ico</Image>

<!-- or -->

<Image height="16" width="16">data:image/x-icon;base64,AAABAAEAEBAAA … DAAA=</Image>

Firefox caches the icon as a [base64](https://en.wikipedia.org/wiki/Base64) data: URI (search plug-ins are stored in the [profile](https://developer.mozilla.org/en-US/docs/Mozilla/Profile_Manager)'s searchplugins/ folder). http: and https: URLs are converted to data: URIs when this is done.

**Note:** For icons loaded remotely (that is, from https:// URIs as opposed to data:URIs), Firefox will reject icons larger than **10 kilobytes**.



**Url**

Describes the URL or URLs to use for the search. The template attribute indicates the base URL for the search query.

Firefox supports three URL types:

* type="text/html" specifies the URL for the actual search query.
* type="application/x-suggestions+json" specifies the URL for fetching search suggestions. In Firefox 63 onwards, type="application/json" is accepted as an alias of this.
* type="application/x-moz-keywordsearch" specifies the URL used when a keyword search is entered in the location bar. This is supported only in Firefox.

For these URL types, you can use {searchTerms} to substitute the search terms entered by the user in the search bar or location bar. Other supported dynamic search parameters are described in [OpenSearch 1.1 parameters](http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_3#OpenSearch_1.1_parameters).

For search suggestions, the application/x-suggestions+json URL template is used to fetch a suggestion list in [JSON](https://developer.mozilla.org/en-US/docs/Glossary/JSON) format. For details on how to implement search suggestion support on a server, see [Supporting search suggestions in search plugins](https://developer.mozilla.org/en/Supporting_search_suggestions_in_search_plugins).

**Param**

The parameters that must be passed in along with the search query as key/value pairs. When specifying values, you can use {searchTerms} to insert the search terms entered by the user in the search bar.

**moz:SearchForm**

The URL for the site's search page for which the plugin. This lets Firefox users visit the web site directly.

**Note:** Since this element is Firefox-specific, and not part of the OpenSearch specification, we use the moz: XML namespace prefix in the example above to ensure that other user agents that don't support this element can safely ignore it.

**Autodiscovery of search plugins**[**Section**](https://developer.mozilla.org/en-US/docs/Web/OpenSearch#Autodiscovery_of_search_plugins)

Web sites with search plugins can advertise them so Firefox users can easily install the plugins.

To support autodiscovery, add a <link> element for each plugin to the <head> of your web page:

<link rel="search"

type="application/opensearchdescription+xml"

title="searchTitle"

href="pluginURL">

Replace the bolded items as explained below:

**searchTitle**

The name of the search to perform, such as "Search MDC" or "Yahoo! Search". This must match your plugin file's <ShortName>.

**pluginURL**

The URL to the XML search plugin, so the browser can download it.

If your site offers multiple search plugins, you can support autodiscovery for them all. For example:

<link rel="search" type="application/opensearchdescription+xml"

title="MySite: By Author" href="http://example.com/mysiteauthor.xml">

<link rel="search" type="application/opensearchdescription+xml"

title="MySite: By Title" href="http://example.com/mysitetitle.xml">

This way, your site can offer plugins to search by author, or by title.

**Supporting automatic updates for OpenSearch plugins**[**Section**](https://developer.mozilla.org/en-US/docs/Web/OpenSearch#Supporting_automatic_updates_for_OpenSearch_plugins)

OpenSearch plugins can automatically update. To support this, include an extra Url element with type="application/opensearchdescription+xml" and rel="self". The templateattribute should be the URL of the OpenSearch document to automatically update to.

For example:

<Url type="application/opensearchdescription+xml"

rel="self"

template="https://example.com/mysearchdescription.xml" />

**Note:** At this time, [addons.mozilla.org](http://addons.mozilla.org/) (AMO) doesn't support automatic updating of OpenSearch plugins. If you want to put your search plugin on AMO, remove the auto-updating feature before submitting it.

**Troubleshooting Tips**[**Section**](https://developer.mozilla.org/en-US/docs/Web/OpenSearch#Troubleshooting_Tips)

If there is a mistake in your Search Plugin XML, you could run into errors when adding a discovered plugin. If the error message isn't be helpful, the following tips could help you find the problem.

* Your server should serve OpenSearch plugins using Content-Type: application/opensearchdescription+xml.
* Be sure that your Search Plugin XML is well formed. You can check by loading the file directly into Firefox. Ampersands (&) in the template URL must be escaped as &amp;, and tags must be closed with a trailing slash or matching end tag.
* The xmlns attribute is important — without it you could get the error message "Firefox could not download the search plugin".
* You **must** include a text/html URL — search plugins including only Atom or [RSS](https://developer.mozilla.org/en/RSS) URL types (which is valid, but Firefox doesn't support) will also generate the "could not download the search plugin" error.
* Remotely fetched favicons must not be larger than 10KB (see [bug 361923](https://bugzilla.mozilla.org/show_bug.cgi?id=361923)).

In addition, the search plugin service provides a logging mechanism that may be useful to plugin developers. Use about:config to set the pref 'browser.search.log' to true. Then, logging information will appear in Firefox's [Error Console](https://developer.mozilla.org/en/Error_Console) (Tools 〉 Error Console) when search plugins are added.