## DNS prefetching

In short, DNS Prefetching is a method of informing the browser of domain names

referenced on a site so that the client can resolve the DNS for those hosts,

cache them, and when it comes time to use them, have a faster turn around on

the request.

### Implicit prefetches

There is a lot of prefetching done for you automatically by the browser. When

the browser encounters an anchor in your html that does not share the same

domain name as the current location the browser requests, from the client OS,

the IP address for this new domain. The client first checks its cache and

then, lacking a cached copy, makes a request from a DNS server. These requests

happen in the background and are not meant to block the rendering of the

page.

The goal of this is that when the foreign IP address is finally needed it will

already be in the client cache and will not block the loading of the foreign

content. Fewer requests result in faster page load times. The perception of this

is increased on a mobile platform where DNS latency can be greater.

#### Disable implicit prefetching

```html

<meta http-equiv="x-dns-prefetch-control" content="off">

```

Even with X-DNS-Prefetch-Control meta tag (or http header) browsers will still

prefetch any explicit dns-prefetch links.

\*\*\_WARNING:\_\*\* THIS MAY MAKE YOUR SITE SLOWER IF YOU RELY ON RESOURCES FROM

FOREIGN DOMAINS.

### Explicit prefetches

Typically the browser only scans the HTML for foreign domains. If you have

resources that are outside of your HTML (a javascript request to a remote

server or a CDN that hosts content that may not be present on every page of

your site, for example) then you can queue up a domain name to be prefetched.

```html

<link rel="dns-prefetch" href="//example.com">

<link rel="dns-prefetch" href="https://ajax.googleapis.com">

```

You can use as many of these as you need, but it's best if they are all

immediately after the [Meta

Charset](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta#attr-charset)

element (which should go right at the top of the `head`), so the browser can

act on them ASAP.

#### Common Prefetch Links

Amazon S3:

```html

<link rel="dns-prefetch" href="//s3.amazonaws.com">

```

Google APIs:

```html

<link rel="dns-prefetch" href="https://ajax.googleapis.com">

```

Microsoft Ajax Content Delivery Network:

```html

<link rel="dns-prefetch" href="//ajax.microsoft.com">

<link rel="dns-prefetch" href="//ajax.aspnetcdn.com">

```

### Further reading about DNS prefetching

\* https://developer.mozilla.org/en-US/docs/Controlling\_DNS\_prefetching

\* https://dev.chromium.org/developers/design-documents/dns-prefetching

\* https://blogs.msdn.microsoft.com/ie/2011/03/17/internet-explorer-9-network-performance-improvements/

\* http://dayofjs.com/videos/22158462/web-browsers\_alex-russel

## Search

### Direct search spiders to your sitemap

After creating a [sitemap](https://www.sitemaps.org/protocol.html)

Submit it to search engine tool:

\* [Google](https://www.google.com/webmasters/tools/sitemap-list)

\* [Bing](https://www.bing.com/toolbox/webmaster)

\* [Yandex](https://webmaster.yandex.com/)

\* [Baidu](https://zhanzhang.baidu.com/)

OR

Insert the following line anywhere in your robots.txt file, specifying the path to your sitemap:

```

Sitemap: https://example.com/sitemap\_location.xml

```

### Hide pages from search engines

According to Heather Champ, former community manager at Flickr, you should not

allow search engines to index your "Contact Us" or "Complaints" page if you

value your sanity. This is an HTML-centric way of achieving that.

```html

<meta name="robots" content="noindex">

```

\*\*\_WARNING:\_\*\* DO NOT INCLUDE ON PAGES THAT SHOULD APPEAR IN SEARCH ENGINES.

### Pingbacks

Your server may be notified when another site links to yours. The href

attribute should contain the location of your pingback service.

```html

<link rel="pingback" href="">

```

\* High-level explanation: https://codex.wordpress.org/Introduction\_to\_Blogging#Pingbacks

\* Step-by-step example case: http://www.hixie.ch/specs/pingback/pingback-1.0#TOC5

\* PHP pingback service: https://web.archive.org/web/20131211032834/http://blog.perplexedlabs.com/2009/07/15/xmlrpc-pingbacks-using-php/

## Social Networks

### Facebook Open Graph data

You can control the information that Facebook and others display when users

share your site. Below are just the most basic data points you might need. For

specific content types (including "website"), see [Facebook's built-in Open

Graph content

templates](https://developers.facebook.com/docs/opengraph/objects/builtin/).

Take full advantage of Facebook's support for complex data and activity by

following the [Open Graph

tutorial](https://developers.facebook.com/docs/opengraph/tutorial/).

For a reference of Open Graph's markup and properties, you may check

[Facebook's Open Graph Protocol reference](http://ogp.me/). Finally,

you can validate your markup with the [Facebook Object

Debugger](https://developers.facebook.com/tools/debug/) (needs

registration to Facebook).

```html

<meta property="fb:app\_id" content="123456789">

<meta property="og:url" content="https://www.example.com/path/to/page.html">

<meta property="og:type" content="website">

<meta property="og:title" content="">

<meta property="og:image" content="https://www.example.com/path/to/image.jpg">

<meta property="og:description" content="">

<meta property="og:site\_name" content="">

<meta property="article:author" content="">

```

### Twitter Cards

Twitter provides a snippet specification that serves a similar purpose to Open

Graph. In fact, Twitter will use Open Graph when Cards is not available. Note

that, as of this writing, Twitter requires that app developers activate Cards

on a per-domain basis. You can read more about the various snippet formats

and application process in the [official Twitter Cards

documentation](https://dev.twitter.com/docs/cards), and you can validate

your markup with the [Card validator](https://cards-dev.twitter.com/validator)

(needs registration to Twitter).

```html

<meta name="twitter:card" content="summary">

<meta name="twitter:site" content="@site\_account">

<meta name="twitter:creator" content="@individual\_account">

<meta name="twitter:url" content="https://www.example.com/path/to/page.html">

<meta name="twitter:title" content="">

<meta name="twitter:description" content="">

<meta name="twitter:image" content="https://www.example.com/path/to/image.jpg">

```

### Google+ / Schema.org

Google also provides a snippet specification that serves a similar

purpose to Facebook's Open Graph or Twitter Cards. While it helps you

to control information displayed on Google+, this metadata is a subset

of [schema.org's microdata vocabulary](https://schema.org/), which

covers many other schemas that can describe the content of your pages

to search engines. For this reason, this metadata is more generic for

SEO, notably for Google's search-engine, although this vocabulary is

also used by Microsoft, Pinterest or Yandex.

You can validate your markup with the [Structured Data Testing

Tool](https://developers.google.com/structured-data/testing-tool/).

Also, please note that this markup requires to add attributes to your

top `html` tag.

```html

<html class="no-js" lang="" itemscope itemtype="http://schema.org/Article">

<head>

<link rel="author" href="">

<link rel="publisher" href="">

<meta itemprop="name" content="">

<meta itemprop="description" content="">

<meta itemprop="image" content="">

```

### Separate mobile URLs

If you use separate URLs for desktop and mobile users, you should consider

helping search engine algorithms better understand the configuration on your

web site.

This can be done by adding the following annotations in your HTML pages:

\* on the desktop page, add the `link rel="alternate"` tag pointing to the

corresponding mobile URL, e.g.:

`<link rel="alternate" media="only screen and (max-width: 640px)" href="https://m.example.com/page.html" >`

\* on the mobile page, add the `link rel="canonical"` tag pointing to the

corresponding desktop URL, e.g.:

`<link rel="canonical" href="https://www.example.com/page.html">`

For more information please see:

\* https://developers.google.com/webmasters/smartphone-sites/details#separateurls

\* https://developers.google.com/webmasters/smartphone-sites/feature-phones