

# What is URL? Structure of the URL

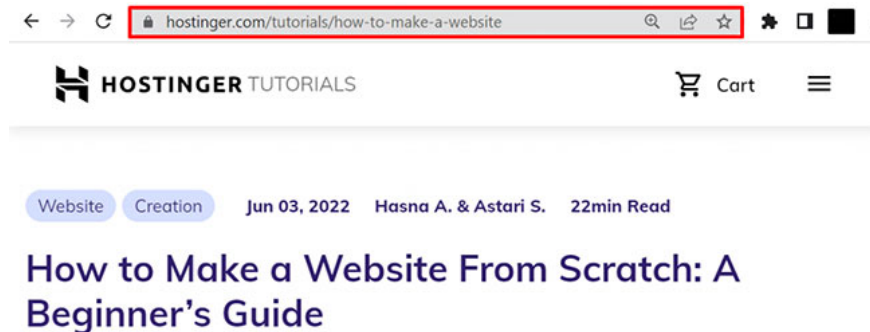
What exactly is a URL, what is its structure and composition? This article will give you an overview of URLs and its structure.

## What is URL?

URL stands for Uniform Resource Locator, translated into Vietnamese as unified resource locator. Specifically, a URL is the address of a unique resource on the Web. Each valid URL will point to a unique resource, that resource can be an HTML page, CSS document, image, video, PDF file. In some exceptional cases, the URL can point to other The resource no longer exists or has been moved to another address (moved).

URLs can contain many different elements. It includes the hostname, which maps to the IP address of a specific resource on the Internet, and a bunch of additional information that tells browsers and servers how to handle things. You can think of the IP address as a phone number, the hostname as the name of the person who owns that phone number you want to look up. And a standard called the Domain Name System (DNS) works in the background like a phone book, translating hostnames into IP addresses that the network uses to route traffic.

## Where is the URL?



A URL can usually be found in the address bar at the top of a web browser's window. On laptops and desktops, the website URL will always show in the address bar as the user scrolls through the website.

In the case of mobile devices, the browser's default behavior causes a URL to disappear as soon as the user starts scrolling down. However, it will reappear when the user scrolls up.

## URL history

Retaining data related to web usage has become a major privacy concern. More and more users are demanding that application service providers and search engines be transparent about the information they collect, retain,

and sell to third parties.

For example, in March 2019, Google updated Chrome's privacy policy. Google notes that in Chrome's basic browser mode, the search engine stores information locally on your system. This information includes browsing history, i.e. the URLs of pages visited, along with caches of text, images, and other resources from those pages.

However, Google also collects and retains data for various periods of time. Some data can be deleted whenever a person wants, some is deleted automatically, and others are retained by Google for longer periods when necessary.

## Structure of the URL

The URL structure was first defined in 1994 by Sir Tim Berners-Lee, the man who created the first web and browser. Essentially, URLs combine domain names with the use of file paths to identify specific file and directory structures. So it's similar to using the C:\Documents\Personal\myfile.txt path in Windows, but adds something on top so it can find the right server on the Internet where that path resides and use the protocol to access the information .

The URL contains several other paths. For example, below is an image of a basic URL, let's analyze its structure.



This simple URL is divided into two main components: Scheme (connection protocol) and Authority (provider).

### Scheme

A lot of people think of a URL as a web address but it's not quite that simple. A web address is a URL but all URLs are not web addresses. Other services you can access on the Internet like FTP or even MAILTO are also URLs. The Scheme part of the URL (the letters following the colon) represents the protocol by which the application (like a web browser) and the server communicate.

Web addresses are the most common URLs, but there are others. Therefore, you will see Schemes like:

1. `http`: This is the underlying protocol of the web, defining what actions web servers and browsers need to take in response to certain commands.
2. Secure HTTP Protocol (`https`): This is a form of HTTP that operates on a secure, encrypted layer for safer transmission of information.
3. File transfer protocol (`ftp`): This protocol is often used to transfer files over the Internet.

HTTP (Hypertext Transfer Protocol) and HTTPS (Hypertext Transfer Protocol Secure) are network communication protocols between web servers and web browsers. They transmit messages, retrieve information through the domain name system (DNS) and send it back to the browser.

The difference between HTTP and HTTPS is that HTTPS encrypts the data transmission. This security protocol better protects your website and is essential for improving your search rankings.

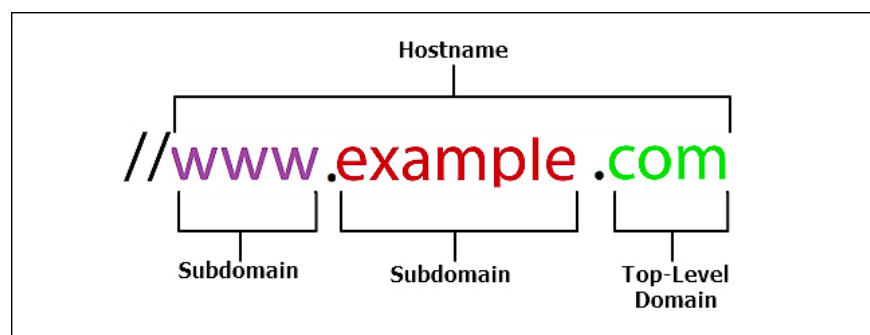
Another difference is that HTTPS uses Transmission Control Protocol/Internet Protocol (TCP/IP) port number 443 encrypted by Transport Layer Security (TLS). Meanwhile, an HTTP URL uses TCP/IP port number 80.

**Important note :** Web browsers can also handle other protocols, including FTP and mailto. FTP allows file sharing between different web servers, locally or remotely. Then direct users to a specific email address.

In modern browsers, Scheme is technically not necessarily part of the URL. If you enter a website like www.TipsMake.com, the browser will automatically determine the appropriate protocol to use. However, some other applications (and protocols) require the use of scheme.

## Authority

The Authority part of a URL (the part that starts after two slashes) is divided into small parts. Start with a simple URL, usually one that takes you to the home page of a website.



In this simple example, the entire 'www.example.com' part is called hostname and is responsible for getting an IP address. If you know the IP address, you can type it into your browser's address bar instead of the hostname.

Here are some of its ingredients:

1. : In the example here, "com" is the top-level domain. This is the highest level in the domain name hierarchy used to translate IP addresses into simple, easy-to-remember language addresses. These top-level domains are created and managed by the Internet Corporation for Assigned Names and Numbers (ICANN). There are three popular top-level domains: .com., .net, .gov. Most countries have two-letter top-level domains, you will see domain names like .us (USA), .vn (Vietnam), .ca (Canada), etc... There are several top-level domains top-level supplements (like .museum) are sponsored and managed by individual organizations. In addition, there are also some shared top-level domains such as .club, .life and .news.
2. **Subdomain** : Subdomain includes any word or phrase that precedes the first period of the URL. Referring to the world wide web, www is the most popular. It indicates that a website is accessible through the Internet and uses HTTP for communication. Because DNS is a hierarchy, both the 'www' and 'example' parts of the example URL above are considered subdomains. The 'www' part is a subdomain of the top-level domain 'com' and the 'www' part is a subdomain of the 'example' domain. That's why you see

companies with registered names like 'google.com' divided into subdomains like 'www.google.com', 'news.google.com', 'mail.google.com', etc... Site owners can use any word as a subdomain for site organization purposes as it points to a specific folder from the main domain. Some of the most popular options are 'blog' and 'news'.

This is the most basic example of the Authority part of a URL, other URLs can be more complex. There are two other components in the Authority section:

1. **User information** : The Authority section may also contain the username and password of the website you are visiting. Nowadays, you are less likely to encounter this URL structure. The user information section comes before the server name and it is followed by an @ sign. For example, you might see a URL that includes user information like this:

```
//username:password@www.example.com
```

1. **Port number** : Network devices use IP addresses to receive information to the appropriate computer on the network. When traffic arrives, the port number tells the computer which application that traffic is targeting. You usually don't see the port number when surfing the web, but you may see it in network applications such as games that require entering a URL. If the URL contains a port number, it appears after the hostname and before a colon. You should see it look something like this:

```
//www.example.com:8080
```

## Additional components of the URL

There are three additional parts of the URL that you can see after the Authority part: path, query, and fragment.

### Path

The Authority part of the URL directs browsers (or other applications) to the correct server on the network. The path (which works like a path in Windows, macOS, or Linux) takes you to the correct folder or file on that server. Paths begin with a slash and have slashes between folders and subfolders as follows:

```
www.example.com/folder/subfolder/filename.html
```

The last part is the file name that will be opened when accessing the website. Even though you might not see this link in the address bar, that doesn't mean it's not there. Some languages used to create web pages hide file names and extensions to make it easier for users to remember and type URLs.

### query

The query portion of a URL is used to identify things that are not part of a fixed path structure. Typically, you'll see them used to perform searches or when websites deliver data through forms. This part of the query is preceded by a question mark and followed by the path (or after the server name if there is no path).

For example, below is the URL when searching for 'wi-fi extender' on Amazon.

```
https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=
```

The search form passed information to Amazon's search engine. The question mark is followed by two parts of the query: the URL for the search (that's the 'url=search-alias%3Daps&field' part) and the entered keyword (that's the 'keywords=wi-fi+extender' part).

This is a pretty simple example, and you'll often see URLs with different plugins. For example, this is the URL when searching for the keyword 'TipsMake' on Google.

<https://www.google.com.vn/search?q=TipsMake&oq=TipsMake&aqs=chrome.69i57j69i60j69>

As you can see, there's some more information here. In this case you will see the browser used.

## Parameters

`html?key1=value1&key2=value2#Some`  
Parameter (*Tham số*)

?key1=value1&key2=value2 are additional parameters provided to the web server. Those parameters are a list of key/value pairs separated by the & symbol. The web server can use those parameters to perform additional work before returning the resource. Each web server has its own rules regarding parameters, and the only reliable way to know whether a particular web server is processing parameters is to ask the web server owner.

## Anchor link (anchor)

`ue2#SomewhereInTheDocument`  
Anchor

#SomewhereInTheDocument is an anchor for another part of the resource itself. An anchor represents a type of "bookmark" within a resource, providing the browser with instructions to display the content located at that "bookmarked" location. For example, on an HTML document, the browser will scroll to the point where the anchor character is identified; on a video or audio document, the browser will try to get to the point the anchor represents. It's worth noting that the part after the # sign, also known as the fragment identifier, is never sent to the server with the request.

## Fragmentation

The last part of the URL that you can see is called the fragment. Fragments are preceded by a pound sign (#) and are used to identify a specific site location. When writing code for a website, designers can create anchor links for specific text such as titles. When using an appropriate fragment at the end of the URL, your browser will load the page and then jump to that anchor link. Anchor links and URLs along with fragmentation are often used to create a website table of contents for easier navigation.

## Types of URLs

In general, the most common types of URLs are absolute and relative.

An absolute URL contains complete information, from protocol to path to resource or parameter. Meanwhile, a relative URL only includes the path to the resource.

## Absolute URLs and Relative URLs

What we saw above is called an absolute URL, but there is also something called a relative URL. The URL standard defines both - although it uses the terms absolute URL strings and relative URL strings, to distinguish them from URL objects (which are in-memory representations of URLs).

Let's look at the difference between "absolute" and "relative" in the context of URLs.

The required parts of a URL depend heavily on the context in which the URL is used. In the browser's address bar, a URL doesn't have any context, so you must provide a full URL (or absolute URL), like the ones we saw above. You do not need to include the protocol (browsers use HTTP by default) or port (only required when the targeted web server is using some unusual port), but all other parts of the URL are necessary.

When a URL is used in a document, such as in an HTML page, things are a little different. Because the browser already has the document's own URL, it can use this information to fill in the missing parts of any URLs available within that document. We can distinguish between absolute URLs and relative URLs by looking at the path part of the URL. If the path part of a URL begins with the character "/", the browser will fetch that resource from the top root of the server without reference to the context provided by the current document.

## Some other types of URLs

Based on functionality, here are some other types of URLs:

1. **Canonical URLs** : Site owners can use them in case they have duplicate content. Canonicalizing a URL is a way to tell search engines which Internet addresses to crawl and index.
2. **Callback URLs** : These refer to the primary destination when a user completes a process on an external system.
3. **Vanity URL** : Also known as custom short URLs, they are easy-to-remember web addresses. Typically, a vanity URL is a redirect of a longer URL. Website owners can use website URL shorteners, such as Bitly, Short.io, and TinyURL, to create a virtual URL.

## Semantic URLs

Although highly technical, URLs represent a human-readable entry point to a website. They can be remembered and anyone can enter them in the browser's address bar. People are the core of the web, and therefore, it is considered best practice to build what are called Semantic URLs. Semantic URLs use words with inherent meanings that anyone can understand, regardless of their technical level.

Linguistic semantics are of course irrelevant to computers. You may often see URLs that look like random combinations of characters. But there are many advantages to creating human-readable URLs:

1. You will have an easier time working with them.
2. Everything will be made clear to everyone, including where they are, what they're doing, what they're reading or interacting with on the web.
3. Some search engines may use such semantics to improve the categorization of linked pages.

Honestly, you can make them however you want. But here are some helpful tips:

1. **There are no special symbols** . Latin letters only, with hyphens (-) instead of spaces. There are no special symbols, no symbols from other languages ??- just simple Latin letters and dashes.
2. **Planning** . Plan your site hierarchy, naming conventions, etc. You shouldn't change anything afterward, so plan carefully. Having to plan everything in advance is also the only major drawback in the entire process of creating semantic URLs.
3. **Don't make the URL too long** : Google recommends no more than 5 words in the page title. As for the overall length of the URL - try to keep it less than 100 characters, including domain.

## URL shortener

URL shortening is a technique in which a URL can be made substantially shorter in length and still point to the requested page. Shorteners do this by using redirects on a short domain name.

There are many URL shortening services. Although many software are free, those that offer capabilities such as web analytics will charge a fee. Companies that offer URL shorteners include Rebrandly, Bitly, Ow.ly, clicky.me, and Budurl.com.

Some web hosting services, such as GoDaddy.com, offer URL shorteners. Other service providers, including search engines, have begun to turn away from URL shorteners because they are often abused by spammers, hiding malware inside URLs. Compact.

## How to use URLs

Any URL can be entered right inside the browser's address bar to access the resource behind it. But this is just the tip of the iceberg!

The HTML language makes extensive use of URLs:

1. to create links to other documents using
2. to link a document to its related resources through various elements such as or

You finished reading the article "**What is URL? Structure of the URL**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.