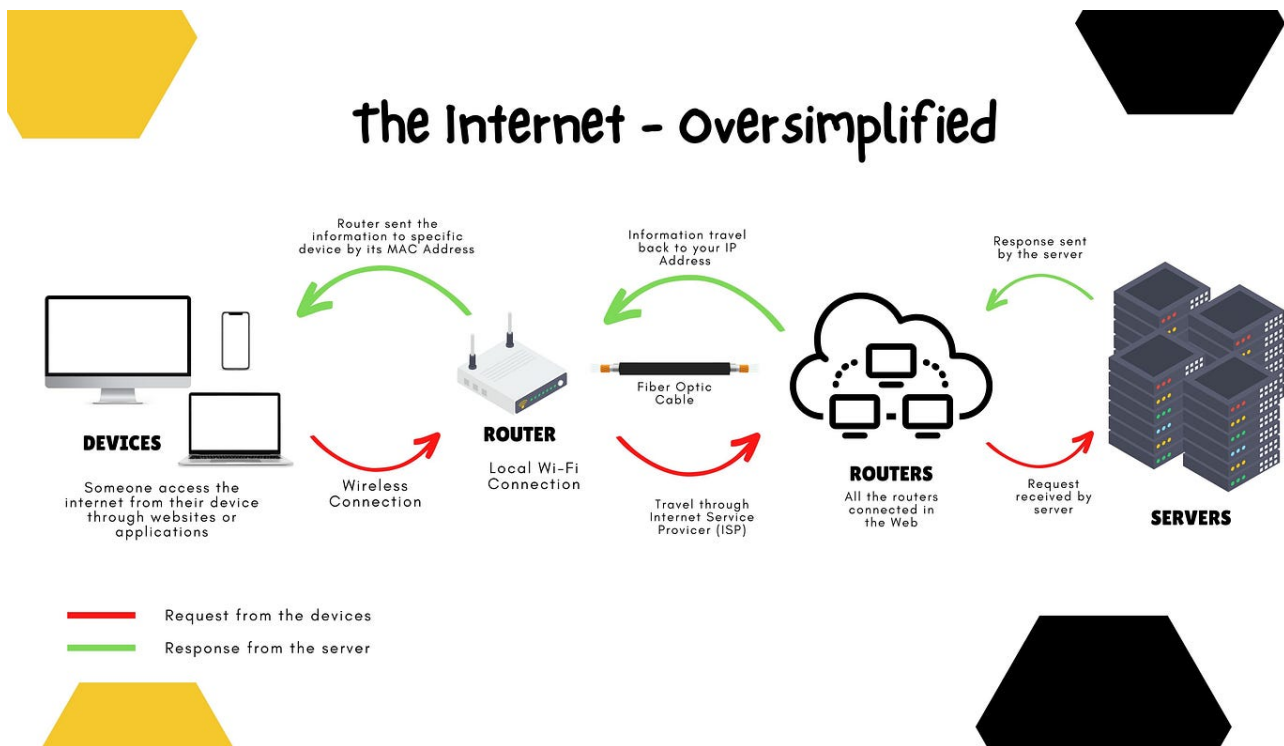


How Web Works ?

Let's start with this picture showcasing a simple anatomy of the working of web.



- At first, someone tried to access a website or application from their devices.
- The request is then forwarded to the gateway of the router to which we have connected wirelessly.
- The router forwards the request to other routers through cables owned by the Internet Service Providers like Airtel, Jio or Vodafone.
- The other routers forwards the request to other routers until they found the target server.
- Once the request is reached to the server, it responds back with the requested data.
- Information is then travel back to our router via the same ISP cables.
- When it reached to our router, it is then forwarded to the specific MAC address of the device who initiated the request in step 1. Thus, completing the journey.

Anatomy of a URL

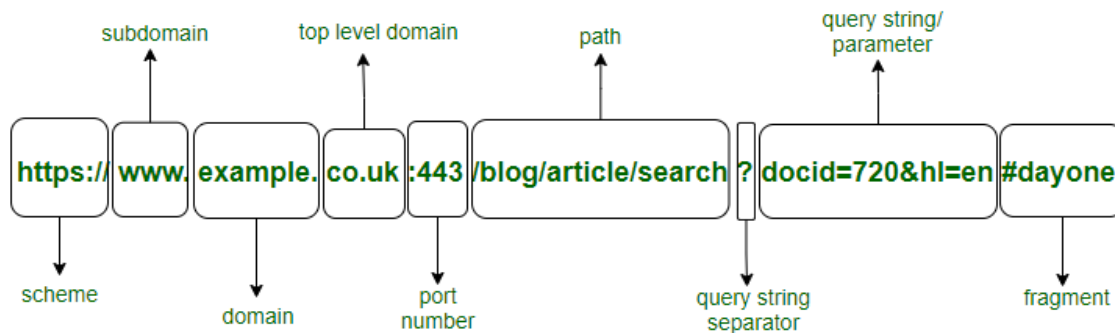
The first step includes sending a request to the server. This is usually done by a web browser where we type a URL address like google.com and the request is thus generated.

<https://t.me/learningnets>

So, let us understand how a URL works by understanding its parts. Here we have a photo that shows different parts of a URL.

Parts of a URL

URL : <https://www.example.co.uk:443/blog/article/search?docid=720&hl=en#dayone>



- At first, we have a scheme - This is the first part of the URL and tells our web browser how to access the website. The most common schemes are "**http://**" and "**https://**". The "**s**" in "**https**" stands for "**secure**" and means the connection is encrypted.
- The Second one is subdomain - This comes after the scheme and is optional. It's like a specific room or section within a website. For example, "blog." or "support." before the main website name.
- The third one is domain - This is the main name of the website, like "example.com". It tells our browser where to find the website on the internet.
- The fourth one is top-level domain - This shows the type of TLD we are using like .com, .gov. , .org, co.uk etc. This also tells about the type of website we are dealing with.
- Next one is port number - Here we can see we are using HTTPS so the port number will be 443. We already knew that from the common ports and protocols section, if you remember
- Moving on, we have the path - This comes after the domain name and tells our browser where to find a specific page or file on the website. It's like giving directions to a room inside a house.
- Next is Query String separator & parameter - This starts with a "?" and contains information to help the website know what specific content to show us. It's like telling the website what we're looking for.
- At last, we have Fragment - This starts with a "#" and takes us to a specific section of the page, like a bookmark. It's like pointing out a specific spot on a page we want to go to.

HTTP Methods

Now that we have generated the request by pressing hard on that Enter button after entering the URL. We will now understand how a HTTP request get transferred to the server using HTTP Methods.

HTTP methods, also known as HTTP verbs, are the actions that a client like a web browser can perform on a resource like a web page that is hosted on a server. There are mainly 4 types of HTTP Methods. These are:

- **GET Request** - This is used for getting information from a web server. It's like asking the server to give you a specific piece of information, like a web page or a list of products
- **POST Request** - This is used for submitting data to the web server and potentially creating new records. It's like filling out a form on a website and submitting it to the server to create a new account or submit a message
- **PUT Request** - This is used for submitting data to a web server to update information. It's like replacing an old version of something with a new one, like updating your profile information on a website.
- **DELETE Request** - This is used for deleting information/records from a web server. It's like deleting something you no longer need, like removing a post you made on a social media platform.

Let see how a sample request flows to the server.

- Open the firefox web browser.
- Press F12.
- Go to network tab.
- Hit reload on any asset
- Show the requests

HTTP Status Codes

HTTP status codes are like a secret language that web servers use to communicate with web browsers. When you visit a website, your browser sends a request to the website's server, and the server responds with a special three-digit number called an HTTP status code.

These codes are divided into five main categories:

- **1xx Informational**: The server is thinking about our request and needs more time to process it.

- **2xx Success:** The server successfully received our request and is sending us the information we have asked for.
- **3xx Redirection:** The server is sending us to a different website or page to get what we need.
- **4xx Client Error:** The server can't understand our request or we don't have permission to access the page.
- **5xx Server Error:** This means, There's a problem on the server side that's preventing it from fulfilling our request.

Some of the most important status codes to know are, like:

- **200 OK:** Everything is working as it should.
- **301 Moved Permanently:** The page we're looking for has moved to a new address.
- **404 Not Found:** This is a famous one that means, The page we have requested doesn't exist on the server.
- **500 Internal Server Error:** There's a problem with the server that's preventing it from fulfilling our request.

HTTP Headers

HTTP headers are like extra information that gets passed between a web browser and a web server when they communicate with each other. They are sent along with the main content (like a web page or an image) to provide additional details about the request or response. Some of the common HTTP headers are:

- **Set-Cookie:** In this, Information is stored which gets sent back to the web server on each request
- **Cache-Control:** It shows the duration to store the content of the response in the browser's cache before it requests it again.
- **Content-Type:** This tells the client what type of data is being returned, i.e., HTML, CSS, JavaScript, Images, PDF or Videos. Using the content-type header the browser then knows how to process the data.
- **Content-Encoding:** It shows the method used to compress the data to make it smaller when sending it over the internet.

Cookies

To understand cookies, you have to understand how HTTP works. So, HTTP is a stateless protocol. That means, HTTP have no feature inherently built into it that can remember and handle requests and sessions.

Think of it like this, let say you went to a shop and the shop owner name is HTTP Kumar. Now this HTTP Kumar has Dementia. Dementia is disease where the affected person suffers from a memory loss. So, you went into the shop, took whatever you wanted to buy and moved out. Now when you reached your car, you remembered that you have left the car keys in the shop itself. You went back to the shopowner HTTP Kumar and told him about your keys. Now, the shop owner have the keys with him however as he is dementia patient, so he forgot who you are and denied your request for your own keys. This happens just within some couple of minutes.

This same things happens with HTTP. You go to the website and you log in. Let's say Instagram. Now, when you try to comment on your crush's new insta reel, a new HTTP POST request is generated for this, however as the HTTP is stateless, it will take it as new request and will ask you to login again. So for every request, you have to authenticate yourself. Sounds like a pain in the ass, isn't it ?

Here comes the sweet cookies into the picture. Cookies are small text files that websites store on your computer or device when you visit them. They help websites remember information about you and your preferences, like your login details, shopping cart contents, or the pages you've visited.

There are different types of cookies:

- **Session cookies** - They are temporary and deleted when you close your browser.
 - **Persistent cookies** - They stay on your device until they expire or you delete them.
 - **First-party cookies** - These are set by the website you're visiting.
 - **Third-party cookies** - These come from other websites, like advertisers, and track your behaviour across multiple sites.
-