What is a Client-Server Architecture?

In this lesson, we will understand the fundamentals of a client-server model, which will help us build a strong foundation for further learning.

We'll cover the following

What is a client?

A *client* is a computer or host that sends the `request` for a specific service or data to a server through the `HTTP` protocol in the form of a `URL` and receives a `response`.

What is a server?

A *server* is a remote machine that accepts the `request`, processes it, and sends the `response` information through the `HTTP/HTTPS` protocol.

> Note: HTTP (Hypertext Transfer Protocol) protocol specifies how the client should send a request to the server and how the server should respond.

> Note: HTTPS (Hypertext Transfer Protocol Secure) encrypts the data before sending it to the client or the server.

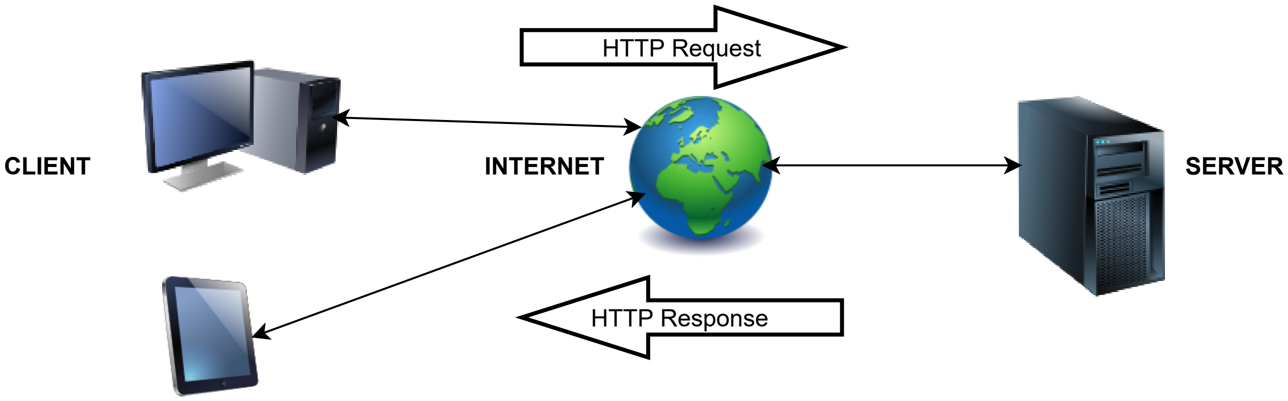
We will discuss and learn more about HTTP and HTTPs in the next few lessons.

What is the client-server model?

When we use AN `Internet` browser to access any web page/website, the *client*, in this case, is the computer running the browser software, which has requested a this case, is the computer running the browser software, which has requested a

web page from a *web server*. The *client* and *server* exchange messages through the `HTTP` `request` and `response` protocol.

The diagram below shows the *client-server* model as discussed above.



Client-Server Architecture

Browser and server interaction flow

> Note: The internet operates on IP addresses. However, we use hostnames, e.g, https://www.educative.io/learn because it's difficult to remember the IP addresses. So, how can we request web pages using a hostname? This is where the DNS (Domain Name System) comes in! It is just like the phonebook of the internet, but it maps the hostname (URL of the website) to its IP address.

When a browser requests for a web page, the following set of events takes place:

1. A user enters the URL of the website or file in the *browser*. The *browser* then requests the `DNS` (Domain Name System) server
2. The `DNS` server looks up the IP address of the *web server* and once it finds it, sends it back to the browser
3. The *browser* then sends over an `HTTP/HTTPS` request to the *web server's* IP (provided by the `DNS` server in the previous step)
4. The *server* responds with the necessary information or data
5. The *browser* renders the data and the website is displayed

In the next lesson, we will learn about API and web services.