Advanced Js

Day 7 - Client Server Model

What is a Client-Server model?

A distributed structure used by computer applications for communication between a sender and receiver.

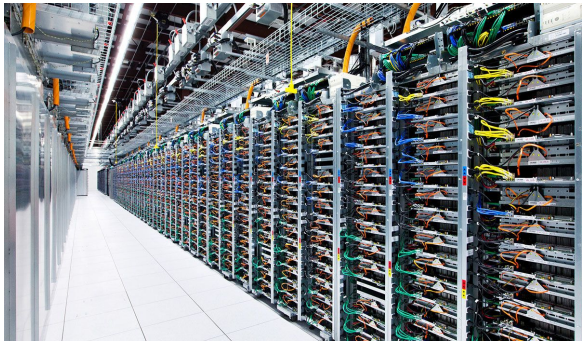
Server is the provider of service and Client is the requester of service.

Client

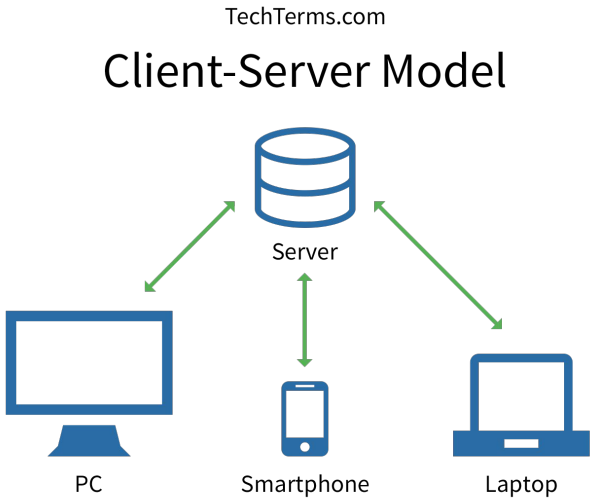
When we talk the word Client, it mean to talk of a person or an organization using a particular service. Similarly in the digital world a Client is a computer (Host) i.e. capable of receiving information or using a particular service from the service providers (Servers).

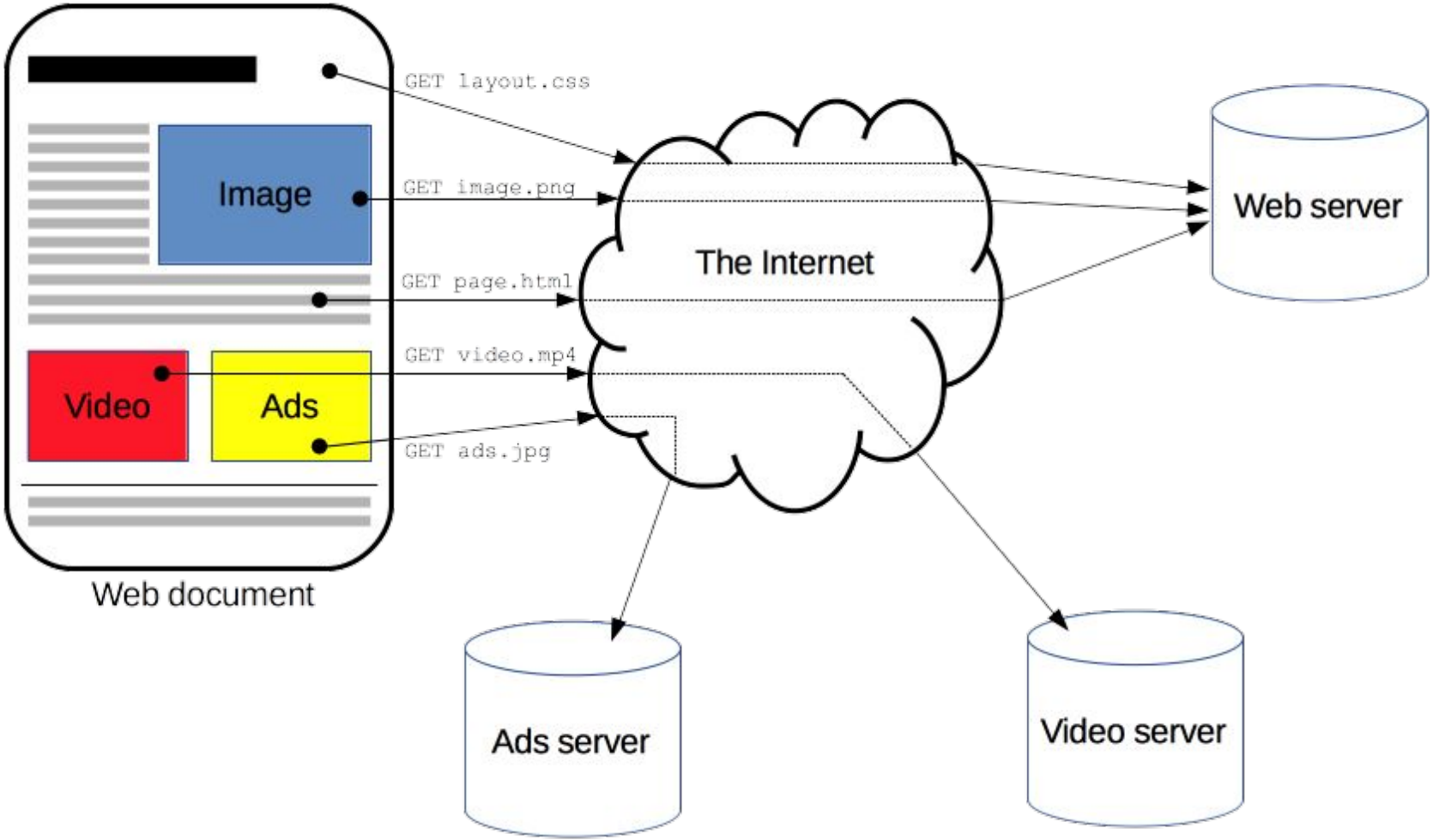
Servers

Similarly, when we talk the word Servers, It mean a person or medium that serves something. Similarly in this digital world a Server is a remote computer which provides information (data) or access to particular services.



Websites use Client-Server model, so how do they work?




GET layout.css
Image
GET image.png
GET page.html
GET video.mp4
Video
Ads
GET ads.jpg
Web document
The Internet
Web server
Ads server
Video server

It’s all about how communication is happening on Internet.

Enter Internet Protocols.

HTTP PROTOCOL

Hypertext Transfer Protocol (HTTP) is a network protocol for transmitting hypermedia documents, such as HTML.

Hypermedia?

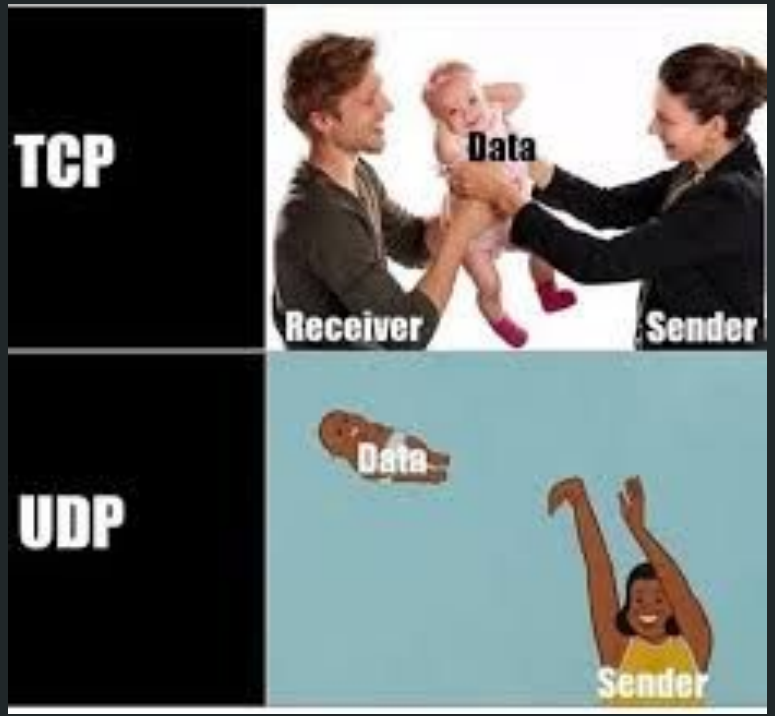
Hypermedia, an extension of the term hypertext, is a nonlinear medium of information that includes graphics, audio, video, plain text and hyperlinks

HTTP allows the fetching of resources, such as HTML documents.

It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser.

More on it later..

TCP/IP PROTOCOL



HTTP, TCP and IP.

HTTP is a protocol used mostly for browsing the internet.

It rides on top of TCP which provides a reliable link between two computers (if packet get lost - it is re-transmitted).

TCP itself rides on top of IP, which provides unified addressing to communicate between computers.

TCP/IP is a basis for internet and 99% of other networks.

The Internet Protocol makes sure the pieces arrive at their destination address.

The TCP protocol can be thought of as the puzzle assembler on the other side who puts the pieces together in the right order, asks for missing pieces to be resent, and lets the sender know the puzzle has been received.

The journey of a website from server to our devices.

Step 1: Enter URL in the browser

Domain Naming System (DNS)

The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses.

DNS translates domain names to IP addresses so browsers can load Internet resources.

https://www.ipvoid.com/

Step 2: Once we reach the servers using URL, Data is ready to be sent to Client. But first...

Transport Layer Security (TLS)

TLS is an encryption protocol designed to secure Internet communication.

A primary use case of TLS is encrypting the communication between web applications (Client) and servers.

TLS Handshake

A TLS connection is initiated using a sequence known as the TLS handshake.

When a user navigates to a website that uses TLS, the TLS handshake begins between the user's device (also known as the *client* device) and the web server.

During a TLS handshake, the two communicating sides exchange messages to acknowledge each other, verify each other, establish the encryption algorithms they will use, and agree on session keys

When does TLS happen?

A TLS handshake takes place whenever a user navigates to a website over HTTPS and the browser first begins to query the website's origin server.

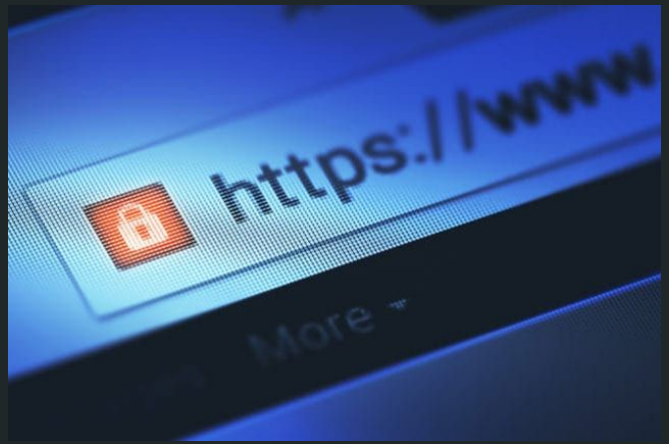
A TLS handshake also happens whenever any other communications use HTTPS, including API calls and DNS over HTTPS queries.

What does TLS do?

- Encryption: hides the data being transferred from third parties.
- Authentication: ensures that the parties exchanging information are who they claim to be.
- Integrity: verifies that the data has not been forged or tampered with.

Once the secure connection is established Data starts flowing from Server to Client.

HTTP



HTTP requests are sent by the browser to server. You can add, update, receive and delete data.

There are various types of HTTP requests to do so and additional information need to be passed to create a request to communicate with the server.

HTTP Headers

HTTP headers let the client and the server pass additional information with an HTTP request or response.

Header types:

- General headers apply to both requests and responses, but with no relation to the data transmitted in the body.
- Request headers contain more information about the resource to be fetched, or about the client requesting the resource.
- Response headers hold additional information about the response, like its location or about the server providing it.
- Entity headers contain information about the body of the resource, like its content length or MIME type.

HTTP Request Methods

GET

The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.

POST

The POST method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server.

PUT

The PUT method replaces all current representations of the target resource with the request payload.

DELETE

The DELETE method deletes the specified resource.

HTTP Response Codes

HTTP response status codes indicate whether a specific HTTP request has been successfully completed.

Responses are grouped in five classes:

1xx: Informational

It means the request was received and the process is continuing.

2xx: Success

It means the action was successfully received, understood, and accepted.

3xx: Redirection

It means further action must be taken in order to complete the request.

4xx: Client Error

It means the request contains incorrect syntax or cannot be fulfilled.

5xx: Server Error

It means the server failed to fulfill an apparently valid request.

200 - OK

201 - Created

202 - Accepted

404 - Not Found

405 - Method Not allowed

408 - Request Timed out

500 - Internal Server Error

503 - Service unavailable

So, How to make HTTP requests to the server?

Curl

curl is used in command lines or scripts to transfer data.

It is also used in cars, television sets, routers, printers, audio equipment, mobile phones, tablets, set top boxes, media players and is the internet transfer backbone for thousands of software applications affecting billions of humans daily.

- Curl head request

curl -I https://www.google.com

- Curl head request with verbose

curl -v -I https://www.google.com

- Curl with explicit http method

curl -X GET https://www.google.com

Postman