



Web Basics I

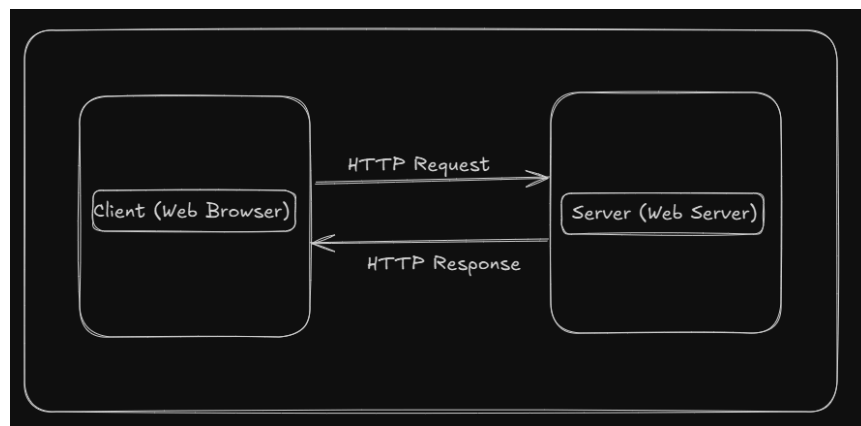
HTTP

Hypertext Transfer Protocol

Protocol → Set of rules [Protocol used to transfer Hypertext]

Hypertext → Text docs → Hyperlink [These hyperlinks connect one document to other documents, resources, or information.]

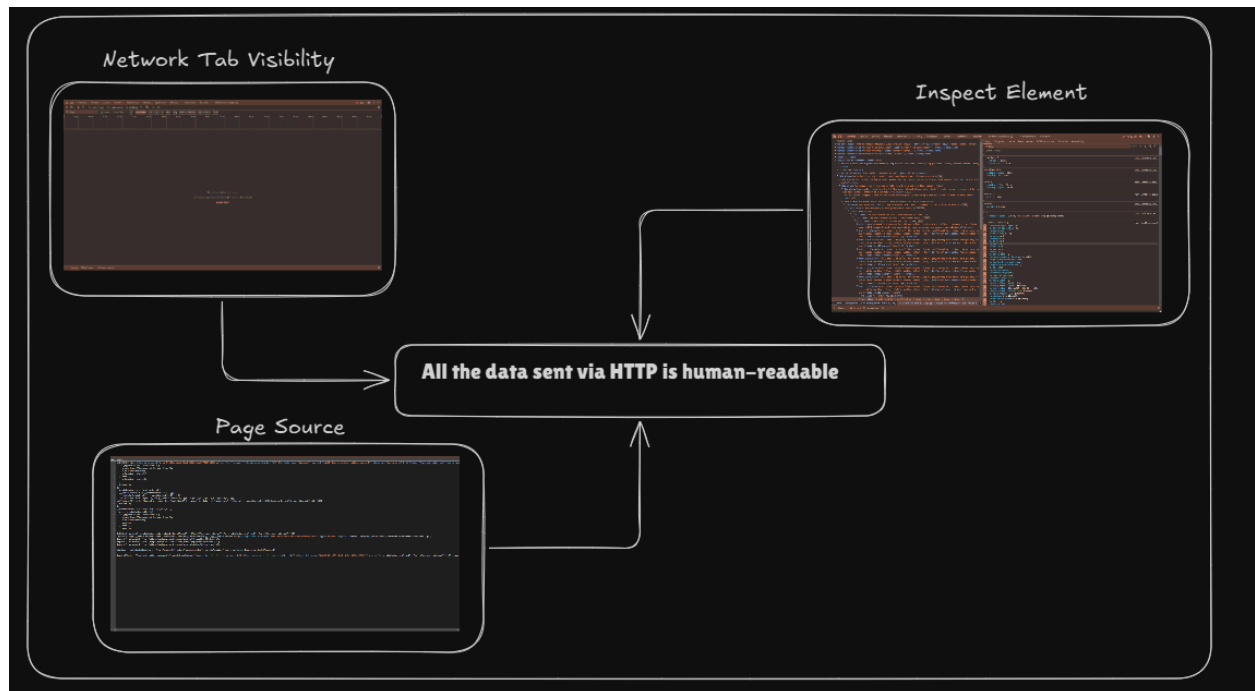
Hypertext Transfer Protocol (HTTP) is a communication protocol used for data transfer between web browsers and servers.



Human Readable

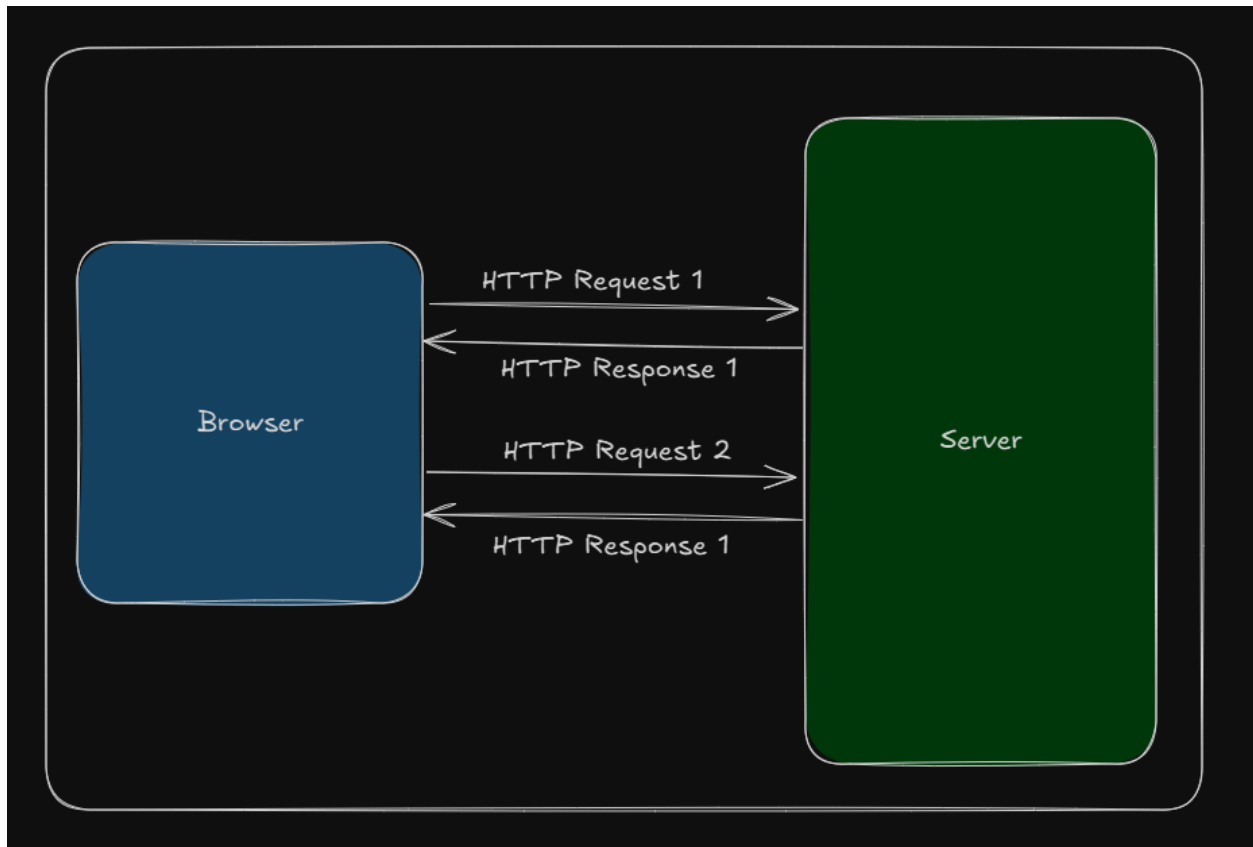
All the data sent via HTTP is human-readable.

- Network tab visibility
- Inspect element
- Page Source



Stateless Protocol

- Every time you visit a page, you are a new user.
- HTTP (Hypertext Transfer Protocol) is a stateless protocol. Each HTTP request is independent, and the server does not have any knowledge of the client's previous requests. If the client sends a request again, the server treats it as a new request.



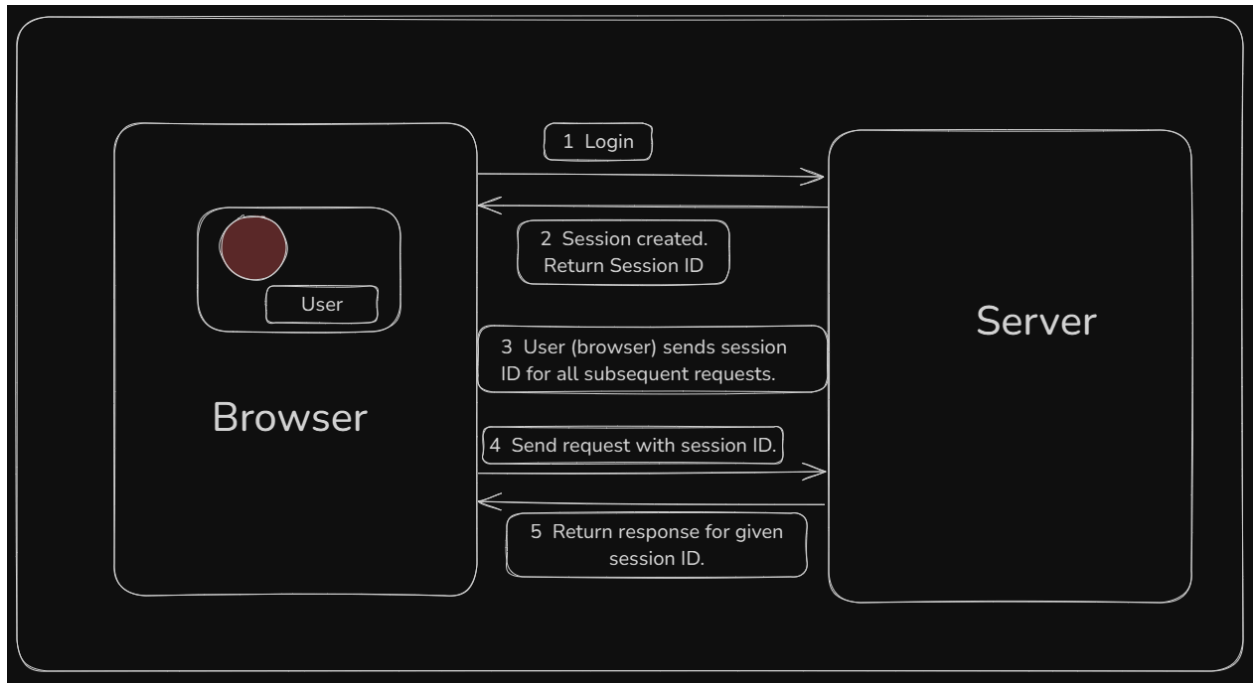
- **Stateless Protocol Does not maintain state or cookies**

Session

Stored state between frontend & backend.

A **session** refers to a period or time frame during which communication is happening between a user and the server, and the server stores some temporary data or state about the user, such as their login status, preferences, or items in the shopping cart.

Example: If you log in to a website, your session starts. The server stores your login details so that as long as you stay on the website, you don't have to log in again. When you log out from the website, the session ends.



When a user starts a session, the server stores some specific information to track the user's activity during the session and provide a seamless experience. Some important aspects of this process are:

1. **Cookies:** Cookies are small files that the server stores in the user's browser. These cookies can store the user's state (like login status, preferences) or session ID. When the user returns to the website, the server identifies their session using the cookies.
2. **Session ID:** The server generates a unique session ID when the user logs in or starts a session. This session ID is sent with each user request to the server. It helps the server know which request is part of which session.
3. **Session Data:** The server also stores temporary data like the user's login status, shopping cart, or preferences. This data is stored as long as the session remains active.

Cookie

Cookies are small files stored in web browsers that are sent by the server to the user's browser.

- State Maintenance

- Storing Login Details
- Personalized Experience

Types of cookies

1. **Session Cookies:** These are temporary cookies that are automatically deleted when the browser is closed. They are used to store session data.
2. **Persistent Cookies:** These cookies remain in the browser until a specific expiration date. They are used to store long-term data, such as maintaining login status for the next visit.

Example: If you are on an online shopping website and fill your shopping cart, the server stores the cart information using cookies. When you visit the website again the next day, your cart remains the same as it was last time.

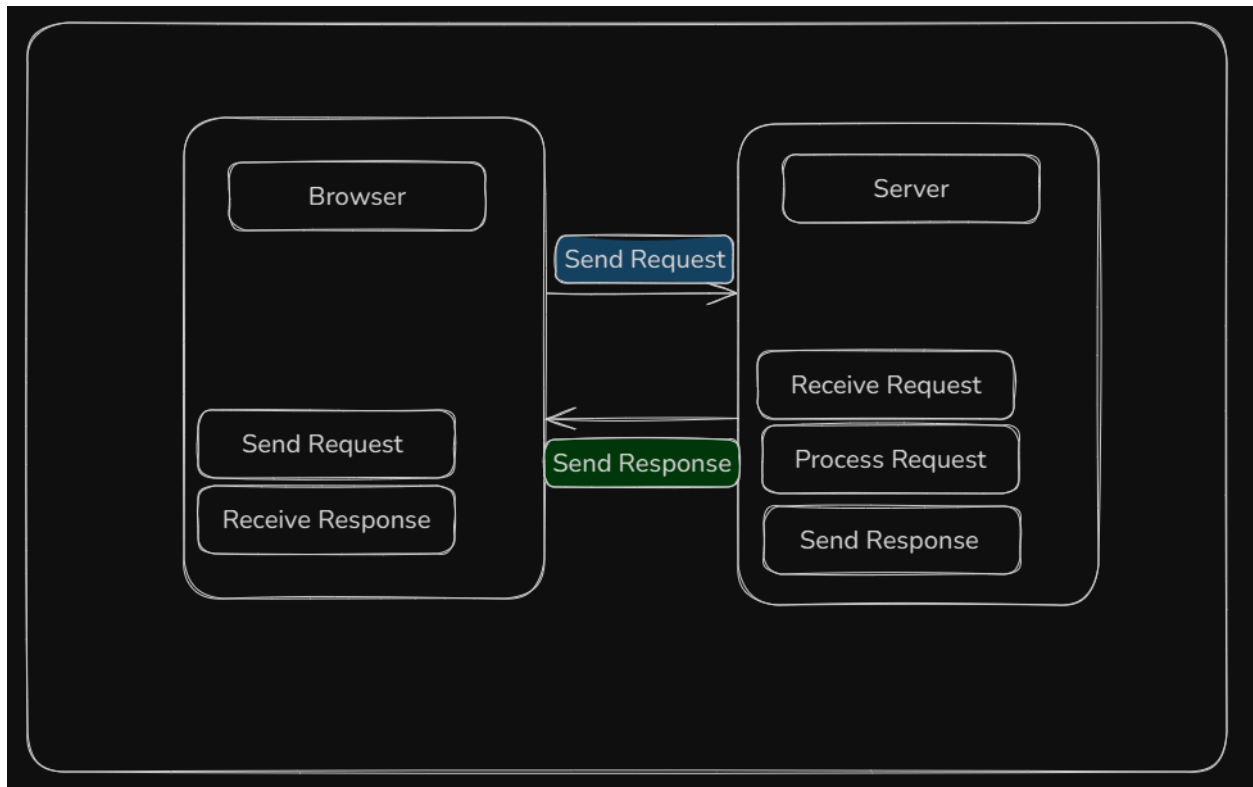
HTTP Headers

- When the browser (client) sends a request to the server, various types of information are sent along with that request, which is in the form of headers.
- HTTP headers provide additional information about the request or response.
- Headers contain metadata that provides extra information about the request or response, such as content type, authentication, cookies, etc.

HTTP headers send information of these types

- Client Information
- Browser Info
- Date Time
- Cookie to store

Request - Response Model



Types of HTTP Requests

1. GET: Retrieve data from the server.
2. POST: Submit data to the server to create or update a resource.
3. PUT: Update or replace an existing resource on the server.
4. PATCH: Updating specific fields of a resource.
5. DELETE: Remove a resource from the server.

Response code

HTTP response codes are status codes sent by the server to inform the client about the result of the request.

Category	Range	Description
1xx	100-199	Informational responses
2xx	200-299	Successful responses
3xx	300-399	Redirection messages

4xx	400-499	Client-side errors
5xx	500-599	Server-side errors

- What action to perform : [GET, POST] Verb
- Where to perform: URL, URI `http://api.sandeep.ai/auth`
- Was it done: `200, 404.....`

HTTP/2

`http/1.1` is a fallback & is still used

HTTP/2 is the second major version of the **Hypertext Transfer Protocol (HTTP)**, designed to improve the performance and efficiency of web communication.

- It was standardized in 2015 as a replacement for HTTP/1.1, focusing on speed, reduced latency, and better use of network resources.

Key Features of HTTP/2

- Use Compression: HTTP/2 uses **HPACK** for compressing headers
- Use Multiplexing [many files at same time]
- Use encryption [`https`]
- Binary Protocol: 0s and 1s

Feature	HTTP/1.1	HTTP/2
Protocol Type	Text-based	Binary-based
Connection	Multiple connections	Single connection
Multiplexing	No	Yes
Header Compression	No	Yes (HPACK)
Server Push	No	Yes
Latency	Higher	Lower

TLS → Transport Layer Security → TLS certificate

`http + TLS = https`

User Agent

User-Agent is a part of the HTTP header, which the client (browser, app, etc.) sends to the server when making a request.

TCP

Transmission Control Protocol

Transmission → Sending

1. Connection-Oriented Protocol

- This means that before sending data, a connection must be established between the sender and the receiver. This connection is established through handshaking.

2. **Data Division into Packets:**

- When you send data (like a file or message), TCP divides that data into smaller packets. These packets are of a particular size, and each packet has its own address and sequence number.

3. **Reliable Data Transfer:**

- Reliability is the most important feature of TCP. When data packets are sent, TCP ensures that all packets reach the receiver in the correct order and in complete form. If any packet is lost or damaged, TCP requests to resend it.

4. **Sequencing:**

- TCP assigns a sequence number to each packet. When the receiver receives the packets, it arranges the data in the correct order based on these numbers to ensure that no data is missed.

5. **Acknowledgment (ACKs):**

- When the receiver successfully receives a packet, it sends an acknowledgment (ACK) to the sender, letting the sender know that the packet has been delivered successfully.
- If the receiver doesn't receive the packet or something goes wrong, it sends a negative acknowledgment (NAK) to the sender, and the sender retransmits

that packet

6. Flow Control:

- TCP uses a flow control mechanism. This comes into play when the sender is sending data too quickly, and the receiver needs time to process that data. This mechanism informs the sender when to stop sending data, preventing the receiver from being overloaded.

7. Congestion Control:

Congestion → traffic

- TCP uses congestion control when there is congestion on the network (e.g., when many devices are sending data simultaneously). It detects congestion and slows down the data transmission to ensure that the network's resources are properly managed.

8. Three-Way Handshake:

To establish a connection, TCP uses a three-way handshake:

- **SYN:** The sender sends a synchronization (SYN) packet to the receiver, requesting a connection.
- **SYN-ACK:** The receiver responds with a synchronization acknowledgment (SYN-ACK) packet, confirming the request and agreeing to the connection.
- **ACK:** The sender sends an acknowledgment (ACK) packet back to the receiver, completing the connection setup.

9. Termination:

- When the data transfer is complete, the connection is gracefully terminated. This is also done through a protocol, in which both the sender and receiver close their connection.

10. Error Checking:

TCP includes a checksum in its packets, which ensures that the packets are transferred without any errors. If a packet is corrupted, the receiver will discard it and request a resend.

FTP

File Transfer Protocol

IP

Internet Protocol

It is a network protocol that defines the method of transferring data between devices.

URL

Uniform Resource Locator

URL → Link → URI → Endpoint

DNS

Domain Name System

Points URL to IP

It is a system that converts human-readable domain names into machine-readable IP addresses on the internet.

Header

Pass additional information

Payload

Actual data → email, password etc.

Cache

Store [temp] the Data

