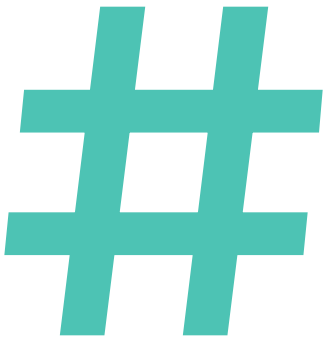

Client and Server

Step 01 Client-Server Architecture

To send a message via the network we can use Client-Server architecture to structure data exchange and resource sharing, this architecture requires two main parts the **client** and the **server**. In a Client-Server architecture, the server responds to the services that are requested by the client.

Step 02 What is a Client?

A client is a software application or a device that initiates communication with a server to request services or resources. In a Client-Server architecture, the client sends (requests) to the server and receives (responses). Clients can be any software that interacts with a server to access data or services such as web browsers, messaging apps, or file transfer programs.

Step 03 What is a Server?

A server is a computer system or software application that provides services or resources to clients over a network. In a Client-Server architecture, the server plays a crucial role by listening to incoming client requests, managing and processing those requests, and sending back the results or responses to the clients.

> NOTE
> Think about the client as a customer in a restorant who sends a request to the server and the server as a chef who receives the request, processes it, and sends the response to the customer.

Step 04 Example of Client and Server

01 Telegram

The client is the Telegram application that you use to send messages, and the server is the Telegram server that receives your messages, processes them, and sends them to the recipient.

02 Google

The client is the browser that you use to search for information, and the server is the Google server that receives your search query, processes it, and sends you the search results.

Step 05 How to Create a Client and Server Application?

To create robust client-server applications that enable efficient data exchange and interaction over a network, we can utilize sockets, which are communication channels that allow two applications to exchange data and interact with each other over a network using the TCP or UDP protocol.