PHP

VivifyAcademy

VIVIFY
ACADEMY

Overview

- Web Server
- Client - Server Communication

Web Server

Server vs. Web Server

A server is a program or device that provides functionality for other programs or devices, called clients. Server is responsible for hosting the website on the internet.

A Web server handles the HTTP protocol. When the Web server receives an HTTP request, it responds with an HTTP response. A Web server is just a “program” that is running on the actual server.

Web servers

Apache HTTP Server: By far the most popular Web server out there

NGINX: Runs on over 30% of all websites.

Apache Tomcat: Open source Java servlet container that functions as a web server.

Node.js: Server-side JavaScript environment for network applications.

Web servers

Built-in PHP Web server:

- Appropriate only for testing purposes.
- php -S localhost:1234
- Runs only one single-threaded process, so PHP applications will stall if a request is blocked.

Web Server links the domain with a folder and serves files from that folder when a request arrives.

Example

- We can set that some domain www.example.com links to some folder /home/user/example-site/
- If we have script script1.php in it, we can execute www.example.com/script1.php
- If we have file image.png in it, we can open the image when we access www.example.com/image.png
- If we have folder subfolder and script script2.php in it, we can execute www.example.com/subfolder/script2.php

Folders and files inside the linked folder are accessible if not explicitly forbidden.

Static content vs. Dynamic content

Files that are considered to be static are files that don't change based on user input, and they consist of things like:

- JavaScript
- CSS
- Images
- HTML files

Static content vs. Dynamic content

Dynamic content is content that requires some kind of processing. For example:

- PHP scripts
- Java files
- C# files

Server-side programming languages

The server-side languages run scripts on a web server. A user's request is fulfilled by running a script directly on the web server to generate some kind of a dynamic response. This response is then sent to the client browser.

- PHP
- Java
- C#
- Python
- NodeJS

Client-side programming languages

The client-side languages run scripts in a browser. The processing takes place on the end users computer. The source code is transferred from the web server to the users computer over the internet and run directly in the browser.

- Javascript

Client - Server Communication

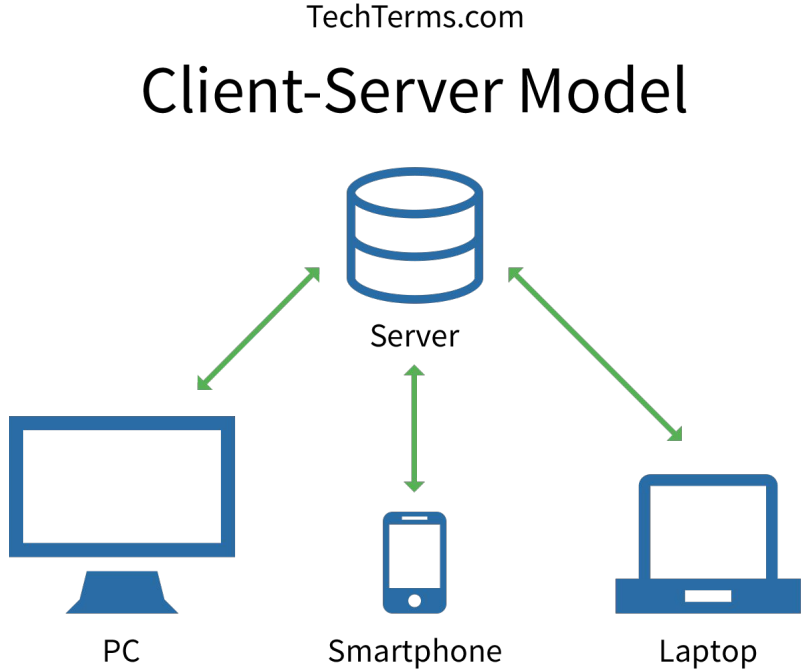
Client - Server Communication

Web servers are what supply the content for web browsers; what the browser requests, the server delivers through Internet network connections.

Web browsers and web servers function together as a **client-server** system.

In computer networking, client-server is a standard method for designing applications where data is kept in central locations (server computers) and efficiently shared with any number of other computers (the clients) on request. All web browsers function as clients that request information from websites (servers).

Client - Server Communication



The other parts of the toolbox

- **Internet connection**: Allows you to send and receive data on the web
- **TCP/IP (Transmission Control Protocol/Internet Protocol)**: Communication protocols that define how data should travel across the web
- **DNS (Domain Name System Servers)**: Address book for websites
- **HTTP (Hypertext Transfer Protocol - application protocol)**: Application protocol that defines a language for clients and servers to speak to each other

Candy Shop Example

Candy Shop Example

- **Internet connection**: like the street between your house and the shop.
- **TCP/IP**: like a car or a bike.
- **DNS**: like looking up the address of the shop so you can access it.
- **HTTP**: like the language you use to order your goods.

So what happens, exactly?

When you type a web address into your browser

1. The browser goes to the **DNS server.**
2. Browser provides the domain name (e.g. www.google.com), and, at the end, it gets the **IP address** of the server where the website is hosted.
3. The browser sends an **HTTP request** message to the server.
4. This message, and all other data sent between the client and the server, is sent across your internet connection using **TCP/IP.**
5. The server process the client's request and sends the **HTTP response.** There are various types of responses, but most common is using **HTML.**
6. The browser interprets the response and shows the result to the user (you see the website as the result)

When you go to http://localhost:8000/vezba1.php

1. Before that you need to start the **web server** by running **php -S localhost:8000.**
2. Browser asks for IP address of the provided domain (**localhost** is domain in this case). Browser first asks the computer where it is installed (your computer). If it can't get the information from the computer, it asks external **DNS providers** for the IP address.
3. In this case, **localhost** domain is recognized by your computer as IP address **127.0.0.1**.
4. This IP address points to your computer. Browser sends **HTTP request** to your computer.
5. **Your computer acts as a server** in this case.

When you go to http://localhost:8000/vezba1.php

1. **:8000** describes which **port** of the server should accept this request.
2. As described in the first step, we started **web server** with **php** command on that port (***php -S localhost:8000***).
3. **Server** is computer, **web server** is a “program” on that computer.
4. Web server process the request (executes PHP script in this case), and returns **HTTP response.**
5. The browser interprets the response and shows the result to the user (you see the result of the executed script in your browser).

Useful links

- https://en.wikipedia.org/wiki/Client%E2%80%93server_model