Architecture client-serveur sur le Web

5TTR Découverte

Objectifs

- Comprendre l'architecture client-serveur du Web
- Identifier le rôle des clients et des serveurs
- Comprendre le fonctionnement du protocole HTTP et HTTPS
- Savoir ce qu'est un port réseau et son utilité

Ce qu'il faut retenir

Le Web repose sur un modèle client-serveur

- Le **client** (souvent un navigateur) envoie une **requête**.
- Le **serveur** reçoit cette requête, effectue un **traitement** (lecture d'un fichier, exécution d'un script...) et renvoie une **réponse**.

👉 Le client **initie toujours** l'échange. 👉 Le serveur **attend** les requêtes.

Qu'est-ce qu'un client ?

- Programme qui envoie des requêtes HTTP.
- Peut être :
  - Un navigateur web (Chrome, Firefox...)
  - Un script (comme `wget`, `curl`, ou `youtube-dl`)
  - Un programme qui récupère des données en ligne
- On parle aussi de **client HTTP** (plus précis que "client web").

Qu’est-ce qu’un serveur ?

- Programme capable de **recevoir une requête HTTP**, de la comprendre, de traiter la demande et d’envoyer une réponse.
- Peut gérer **plusieurs types de services** via différents **ports réseau**.
- Exemples de ports :
  - **80** pour HTTP
  - **443** pour HTTPS
  - D’autres ports existent pour d’autres protocoles (ex: FTP, SMTP…)

À quoi sert un port ?

Un port est comme une **porte d’entrée** spécifique d’un serveur. Il permet de diriger la requête vers le bon service.

📞 L’analogie du central téléphonique : Comme dans une entreprise où tu fais « tapez 1 pour la compta, tapez 2 pour les ventes… », le client web contacte un port précis pour parler au bon service (programme).

HTTP vs HTTPS

HTTP	HTTPS
Non sécurisé	Sécurisé par chiffrement (via TLS)
Port 80	Port 443
Données visibles	Données chiffrées
Suffisant en local	Obligatoire en production

🔒 HTTPS ajoute une couche de **sécurité et d’authentification**. Il est de plus en plus utilisé (souvent favorisé par Google).

À retenir en 5 points

1. Le web repose sur un modèle **client-serveur** où le client envoie une requête, et le serveur y répond.
2. Le **client HTTP** est un programme qui envoie des requêtes (navigateur, script, etc.).
3. Le **serveur HTTP** est un programme qui écoute sur un port et traite les requêtes reçues.
4. Le **port** permet de cibler le bon service sur un serveur (HTTP = 80, HTTPS = 443).

5. **HTTPS** est la version sécurisée de HTTP, utilisée pour protéger les données et vérifier l'identité du serveur.

Si tu veux héberger un site localement, tu utiliseras un **serveur HTTP local** comme Apache ou Nginx, via un package facile à installer (ex: Laragon, XAMPP).

© 2025 Mr. Lambrechts
https://v2.ttrinfo.be
3 | 3