

Interface Homme Machine sur le Web

02 Avril 2024

1 Client - Serveur

Deux ordinateurs en réseau peuvent s'échanger des données. Dans la plupart des cas ces échanges ne sont pas "symétriques" : C'est à dire qu'un ordinateur A va souvent se contenter de demander des ressources (fichiers contenant du texte, photos, vidéos, sons...) à un ordinateur B. De l'autre côté, l'ordinateur B va lui se contenter de fournir des ressources à tous les ordinateurs qui lui en feront la demande. On dira alors que l'ordinateur A est un client alors que l'ordinateur B est un serveur.

En tapant «<http://www.google.fr>» dans votre navigateur, votre machine va chercher à entrer en communication avec le serveur portant le nom «www.google.fr». Une fois la liaison établie, le client et le serveur vont échanger des informations en dialoguant :

2 Exemple

client : bonjour www.google.fr, pourrais-tu m'envoyer le fichier `index.html`

serveur : OK client, voici le fichier `index.html`

client : je constate que des images et du code css sont utilisés, envoies les moi.

serveur : OK, les voici

Évidemment ce dialogue est très imagé.

3 Sur internet

Sur internet, ce modèle client/serveur domine assez largement, même s'il existe des cas où un ordinateur pourra jouer tour à tour le rôle de client et le rôle de serveur, très souvent, des ordinateurs (les clients) passeront leur temps à demander des ressources à d'autres ordinateurs (les serveurs) . Par exemple, comme expliqué dans l'exemple ci-dessus on retrouve cet échange client/serveur à chaque fois que l'on visite une page web. Il y a de fortes chances pour que votre ordinateur personnel joue quasi exclusivement le rôle de client.

N'importe quel type d'ordinateur peut jouer le rôle de serveur, mais dans le monde professionnel les serveurs sont des machines spécialisées conçues pour fonctionner 24h sur 24h. Ils peuvent aussi avoir une grosse capacité de stockage afin de stocker un grand nombre de ressources (vidéos, sons,...).

Souvent les serveurs sont spécialisés dans certaines tâches, par exemple, les serveurs qui envoient aux clients des pages au format HTML sont appelés "serveur web".

4 Web statique et Web dynamique

Il y a quelques années, le web était dit « statique » : le concepteur de site web écrivait son code HTML et ce code était simplement envoyé par le serveur web au client. Les personnes qui consultaient le site avaient toutes le droit à la même page, le web était purement « consultatif ».

Les choses ont ensuite évolué : les serveurs sont aujourd'hui capables de générer eux-mêmes du code HTML. Les résultats qui s'afficheront à l'écran dépendront donc des demandes effectuées par l'utilisateur du site : le web est devenu dynamique.

Différents langages de programmation peuvent être utilisés « côté serveur » afin de permettre au serveur de générer lui-même le code HTML à envoyer. Le plus utilisé encore aujourd'hui se nomme PHP. Nous n'étudierons pas en profondeur le langage PHP, néanmoins voici un exemple de code permettant d'afficher l'heure en direct.

```
1  <?php
2      $heure = date("H:i");
3      echo '<h1>Premiere NSI</h1>
4          <p>Nous sommes en cours, il est : '.$heure.'</p>';
5  ?>
```

Lorsqu'un client viendra à se connecter à la page web, celui ci enverra une demande au serveur web qui exécutera le code et lui enverra le code HTML ci-dessous :

```
1  <h1>Premiere NSI</h1>
2  <p>Nous sommes en cours, il est : 11h25</p>
```

5 Protocole et requête HTTP

Revenons sur l'adresse qui s'affiche dans la barre d'adresse d'un navigateur web et plus précisément sur le début de cette adresse c'est-à-dire le "http".

Le protocole (un protocole est ensemble de règles qui permettent à 2 ordinateurs de communiquer ensemble) HTTP (HyperText Transfert Protocol) va permettre au client d'effectuer des requêtes à destination d'un serveur web. En retour, le serveur web va envoyer une réponse.

Voici une version simplifiée de la composition d'une requête HTTP (client vers serveur) :

- la méthode employée pour effectuer la requête
- l'URL de la ressource
- la version du protocole utilisé par le client (souvent HTTP 1.1)
- le navigateur employé (Firefox, Chrome) et sa version
- le type du document demandé (par exemple HTML)

Question : Essayer de trouver ces informations lorsque vous allez sur le site : <http://www.lycee-bichat.fr/>

6 Méthodes de requête HTTP

Une requête HTTP utilise une méthode (c'est une commande qui demande au serveur d'effectuer une certaine action). Voici la liste des méthodes disponibles :

GET, HEAD, POST, OPTIONS, CONNECT, TRACE, PUT, PATCH, DELETE

Détaillons 4 de ces méthodes :

- GET : C'est la méthode la plus courante pour demander une ressource. Elle est sans effet sur la ressource.
- POST : Cette méthode est utilisée pour soumettre des données en vue d'un traitement (côté serveur). Typiquement c'est la méthode employée lorsque l'on envoie au serveur les données issues d'un formulaire.
- DELETE : Cette méthode permet de supprimer une ressource sur le serveur.
- PUT : Cette méthode permet de modifier une ressource sur le serveur

7 Réponse du serveur

Une fois la requête reçue, le serveur va renvoyer une réponse, cela peut varier selon le navigateur, le serveur, etc..

Voici un exemple de réponse du serveur :

```
1 HTTP/1.1 200 OK
2 Date: Thu, 15 feb 2024 12:02:32 GMT
3 Server: Apache/2.0.54 (Debian GNU/Linux) DAV/2 SVN/1.1.4
4 Connection: close
5 Transfer-Encoding: chunked
```

Plusieurs informations vont être envoyées comme l'heure, la version du protocole mais surtout le code de validation.

Sur la première ligne nous pouvons voir le code 200 indiquant que le document recherché par le client a bien été trouvé par le serveur. Il existe d'autres codes dont un que vous connaissez peut-être déjà.

8 Protocole HTTPS

Le "HTTPS" est la version "sécurisée" du protocole HTTP. Par "sécurisé" on entend que les données sont chiffrées avant d'être transmises sur le réseau.

- le client demande au serveur une connexion sécurisée (en utilisant "https" à la place de "http" dans la barre d'adresse du navigateur web)
- le serveur répond au client qu'il est OK pour l'établissement d'une connexion sécurisée. Afin de prouver au client qu'il est bien celui qu'il prétend être, le serveur fournit au client un certificat prouvant son "identité".
- les échanges entre le client et le serveur seront chiffrés grâce à un système de clé de chiffrement.

D'un point vu pratique il est nécessaire de bien vérifier que le protocole est bien utilisé (l'adresse commence par "https") avant de transmettre des données sensibles (coordonnées bancaires...). Si ce n'est pas le cas, toute personne qui interceptera les paquets de données sera en mesure de lire vos données sensibles.