

06. Principe du modèle client-serveur



Description de l'activité

Dans cette activité, on présente les grands principes du modèle client-serveur. On montre en particulier un exemple de dialogue simple entre un client et un serveur et on aborde la notion de pages dynamiques.

Objectifs pédagogiques ou compétences

Objectifs généraux	Objectifs intermédiaires	Compétences
Connaître le principe du modèle client-serveur	Comprendre ce qu'est un serveur web Comprendre ce qu'est une page web dynamique	Lire un texte et en synthétiser les informations utiles Schématiser un fonctionnement simple client-serveur

Tags

#web # modèleClient-serveur #pagesDynamiques # serveurWeb

Matériel

- Une fiche activité Etude de document par élève

REMARQUE :

CETTE ACTIVITE PEUT ETRE REALISEE EN INTRODUCTION A L'ACTIVITE 05. URL, PROTOCOLE HTTP ET MODELE CLIENT-SERVEUR

Déroulé de l'activité

Introduction : (5 minutes)

Pour introduire la notion, on pose la question « **Qu'est-ce qu'un serveur ?** » à la classe entière. Il s'agit d'amorcer un dialogue avec les élèves et d'évaluer leurs connaissances sur le sujet.

Étape 1 – Etude de document : (25 minutes)

L'enseignant.e fourni le document présenté sur la fiche activité élèves – ETUDE DE DOCUMENT :

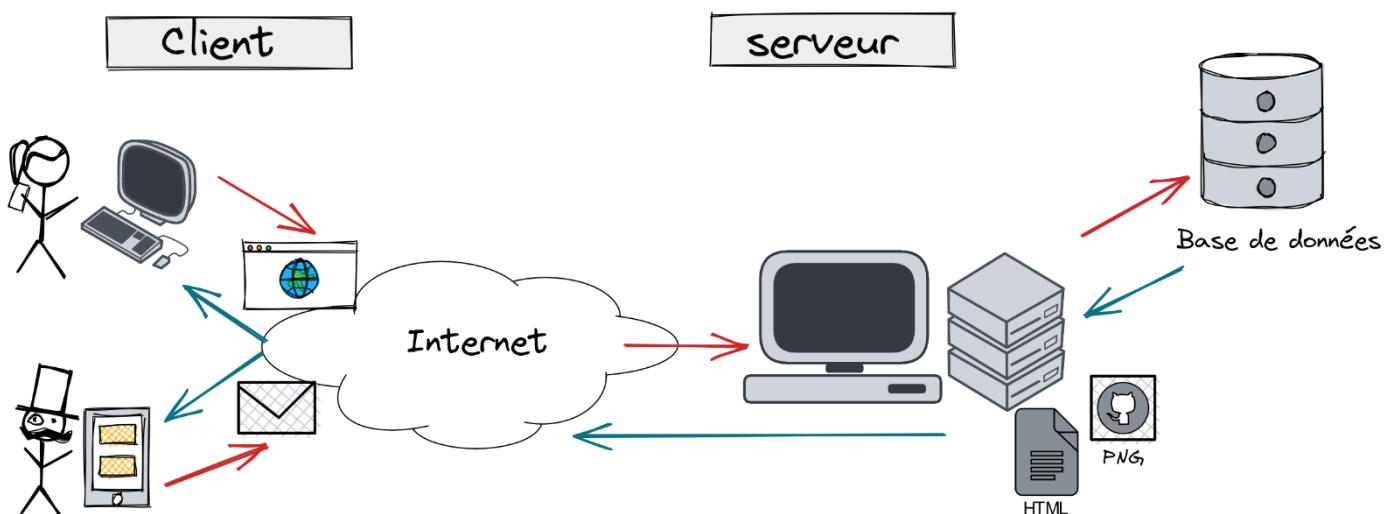
En individuel, les élèves disposent de 25 minutes pour lire le texte et répondre aux questions.

Correction des questions :

1. Qu'est-ce que la relation client-serveur ?

La relation client-serveur est un modèle de communication informatique qui définit comment les ordinateurs, les applications ou les dispositifs interagissent les uns avec les autres au sein d'un réseau ou d'un système distribué. Cette relation est basée sur un modèle de communication où un composant, appelé le client, sollicite des services ou des informations auprès d'un autre composant, appelé le serveur.

2. En un schéma simple, expliquez le fonctionnement client-serveur pour afficher une page web.



3. Qu'est-ce qu'une page dynamique sur le web ?

Une page dynamique sur le web est une page web dont le contenu peut changer ou s'adapter en temps réel en fonction des interactions de l'utilisateur, des données en cours de traitement ou d'autres facteurs. Contrairement aux pages statiques qui affichent un contenu fixe et prédéfini, les pages dynamiques offrent une expérience utilisateur plus interactive et personnalisée.

Étape 2 : Synthèse et conclusion (15 minutes)

L'enseignant.e propose une correction collective des questions de l'étude de document, et s'assure que les notions sont bien comprises.

L'enseignant.e peut aussi ouvrir sur les métiers en lien avec la gestion de client-serveur.

Quelques exemples de métiers par domaines :

Informatique et Technologies de l'Information :

- Administrateur.trices systèmes : gèrent des réseaux et des serveurs pour garantir le bon fonctionnement des systèmes client-serveur.
- Ingénieur.es en réseau : conçoivent, développent et maintiennent des infrastructures de réseau pour prendre en charge la communication entre clients et serveurs.
- Développeur.euses de logiciels : créent des applications clientes et serveurs pour diverses industries et besoins.

Systemes d'entreprise :

- Administrateur.rices de bases de données : gèrent des serveurs de bases de données pour stocker et gérer des données métier critiques.
- Administrateur.rices de serveurs : assurent le fonctionnement, la sécurité et la maintenance des serveurs informatiques dans les entreprises.
- Analystes en sécurité informatique : veillent à ce que les systèmes client-serveur soient sécurisés contre les menaces potentielles.

Télécommunications :

- Opérateur.rices de réseaux mobiles : gèrent les réseaux mobiles, où les smartphones agissent en tant que clients se connectant aux serveurs pour les appels, les messages et les données.

Fiche activité élèves

Etude de document - (25 minutes)

Nous avons vu que deux ordinateurs en réseau peuvent s'échanger des données. Dans la plupart des cas ces échanges ne sont pas "symétriques" : en effet un ordinateur **A** va souvent se contenter de demander des ressources (fichiers contenant du texte, photos, vidéos, sons...) à un ordinateur **B**. 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** (celui qui demande des ressources) est un client alors que l'ordinateur **B** (celui qui fournit les ressources) sera qualifié de serveur.

En tapant «<http://www.google.fr>», votre machine va chercher à entrer en communication avec le serveur portant le nom «www.google.fr» (en fait c'est plus compliqué, pour les puristes nous dirons donc que la communication va être établie avec le serveur [www](http://www.google.fr) du domaine [google.fr](http://www.google.fr), mais bon, pour la suite nous pourrions nous contenter de l'explication « simplifiée »).

Une fois la liaison établie, le client et le serveur vont échanger des informations en dialoguant :

Client : bonjour www.google.fr (ou bonjour [www](http://www.google.fr) se trouvant dans le domaine [google.fr](http://www.google.fr)), pourrais-tu m'envoyer le fichier [index.html](http://www.google.fr/index.html) ?

Serveur : OK client, voici le fichier [index.html](http://www.google.fr/index.html)

Client : je constate que des images, du code css sont utilisés, peux-tu me les envoyer ?

Serveur : OK, les voici

Évidemment ce dialogue est très imagé, mais il porte tout de même une part de « vérité ».

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 (exemple le "peer to peer", voir le module "Internet"). 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 (sauf si vous êtes un adepte du "peer to peer").

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,...).



Serveur

Afin d'assurer une continuité de service, dans les sociétés, plusieurs serveurs assurent exactement le même rôle (on parle de redondance). Vous vous doutez bien que Google ne possède pas qu'un seul serveur. En effet, en moyenne, chaque seconde, c'est environ 65 000 clients qui se connectent aux serveurs du moteur de recherche de Google. Aucun serveur, même extrêmement performant, ne serait capable de répondre à toutes ces requêtes.

Google, Amazon ou encore Facebook possèdent un très grand nombre de serveurs afin de pouvoir satisfaire les demandes des utilisateurs en permanence. Ces entreprises possèdent d'immenses salles contenant chacune des centaines ou des milliers de serveurs (ces serveurs sont rangés dans des armoires appelées "baie serveur").



Salle serveur

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".

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. D'autres langages sont utilisables côté serveur (pour permettre la génération dynamique de code HTML) : Java, Python...

Voici un exemple très simple de code en PHP :

```
<?php
$heure = date("H:i");
echo '<h1>Bienvenue sur mon site</h1>
      <p>Il est ', $heure, '</p>';
?>
```

Sans entrer dans les détails, si un client se connecte à un serveur web qui exécute ce code à 18h23, le serveur enverra au client le code HTML ci-dessous :

```
<h1>Bienvenue sur mon site</h1>
<p>Il est 18h23</p>
```

En revanche si un client se connecte à ce même serveur à 9h12, le serveur enverra au client le code HTML ci-dessous :

```
<h1>Bienvenue sur mon site</h1>
<p>Il est 9h12</p>
```

Comme vous pouvez le constater, le PHP permet de générer des pages HTML dynamiquement. Inutile de préciser que cet exemple est volontairement très simple, le PHP est capable de générer des pages HTML bien plus complexes.

Questions

Après avoir lu le texte ci-dessus, répondez aux questions suivantes :

1. Qu'est-ce que la relation client-serveur ?

.....
.....
.....
.....

2. En un schéma simple, expliquez le fonctionnement client-serveur pour afficher une page web.

3. Qu'est-ce qu'une page dynamique sur le web ?

.....
.....
.....
.....