

Modèle Client Serveur Web – Requête HTTP

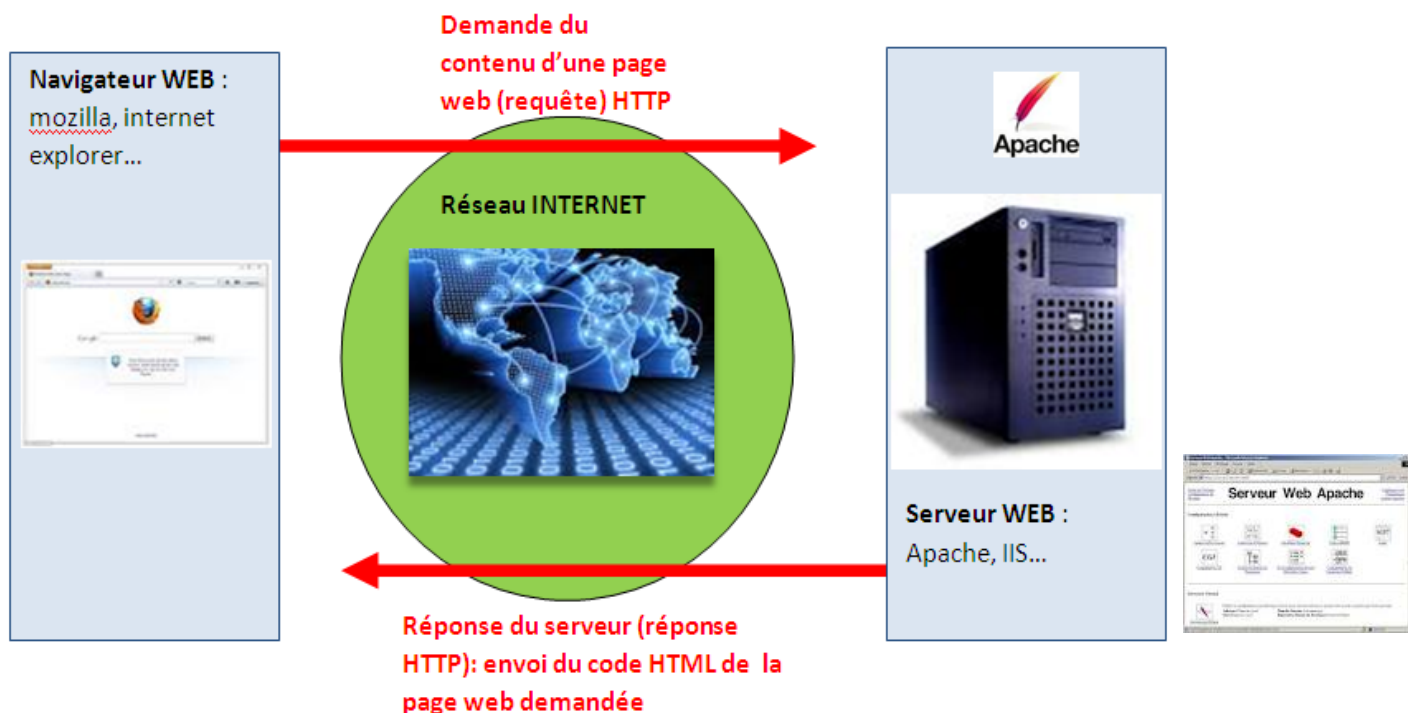
1. Modèle Client-Serveur statique

On a deux acteurs principaux dans une communication Web : **Les navigateurs et les serveurs web.**

Les serveurs WEB hébergent les pages les pages HTML.

Un serveur WEB stocke sur son disque dur le contenu d'un site internet. Ce serveur WEB est identifié sur le réseau Internet par un identifiant unique : son adresse IP (exemple d'adresse IP: 215.56.147.3).

Les navigateurs demandent le code HTML des pages web aux serveurs qui répondent aux demandes des navigateurs via le protocole http (hypertext transfert protocol).

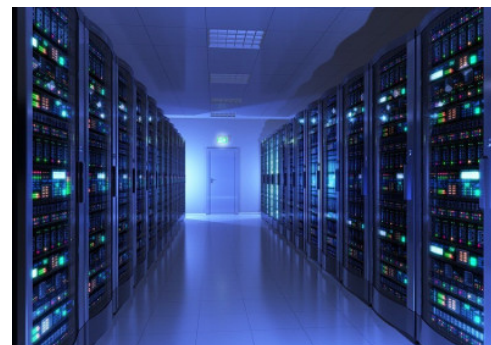


Remarque :

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,...).



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 65000 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"**).



2. Le protocole 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"

Selon les cas cette adresse commencera par http ou https *.

Le protocole 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**.

* 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. Seul le possesseur de la clé de déchiffrement sera en mesure de lire les données transmises sur le réseau. D'un point de vue strictement 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, passez votre chemin, car toute personne qui interceptera les paquets de données sera en mesure de lire vos données sensibles.

• Voici un exemple de requête HTTP :

```
GET /mondossier/monFichier.html HTTP/1.1
User-Agent : Mozilla/5.0
Accept : text/html
```

Nous avons ici plusieurs informations :

- "GET" est la méthode employée (voir ci-dessous)
- "/mondossier/monFichier.html" correspond l'URL de la ressource demandée
- "HTTP/1.1" : la version du protocole est la 1.1
- "Mozilla/5.0" : le navigateur web employé est Firefox de la société Mozilla
- "text/html" : le client s'attend à recevoir du HTML

Revenons sur la méthode employée :

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

• Réponse du serveur à une requête HTTP

Une fois la requête reçue, le serveur va renvoyer une réponse, voici un exemple de réponse du serveur :

```
HTTP/1.1 200 OK
Content-type: text/html

<html >
  <head>
    <meta charset="utf-8">
    <title>Voici mon site</title>
  </head>
  <body>
    <h1>Hello World! Ceci est un titre</h1>
```

La 1re ligne se nomme la ligne de statut :

HTTP/1.1 : version de HTTP utilisé par le serveur

200 : code 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 : le code 404 (qui signifie «Le document recherché n'a pu être trouvé»).

3. Simulation requête HTTP

• À faire vous-même 1

En utilisant le simulateur de réseau Filius et en vous aidant de la vidéo ci-dessous :

https://www.youtube.com/watch?v=EZp_TLGVyv0

Mettez en place un serveur web dans le réseau proposé dans le fichier [snt_sim_http.flr](#) (fichier à ouvrir depuis le logiciel Filius). **Copiez** cette page HTML dans le fichier index du serveurWeb :

```
<html >
  <head>
    <meta charset="utf-8">
    <title>Voici mon site</title>
  </head>
  <body>
    <h1>Hello World! Ceci est un titre</h1>
    <p>Ceci est un <strong>paragraphe</strong>. Avez-vous bien compris ?</p>
  </body>
</html>
```

Testez ce serveur web en utilisant un navigateur web sur le client de votre choix (M1, M2,...).

À noter : afin d'étudier les échanges de données entre un client et un serveur, nous utilisons l'outil "data exchange", cet outil fournit énormément d'informations, dans la vidéo ci-dessus nous nous intéressons uniquement aux requêtes HTTP mais "data exchange" propose aussi d'étudier les protocoles IP et TCP (voir le module "internet").

• À faire vous-même 2

En vous aidant de ce qui a été fait dans le module "Internet", **mettez** en place un serveur DNS dans le réseau créé au "À faire vous-même 1" afin de faciliter la consultation du site web. Vous **désignerez** comme nom de domaine : SNT

