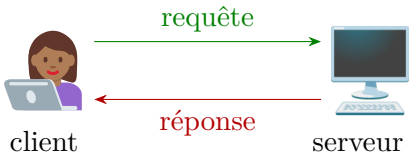
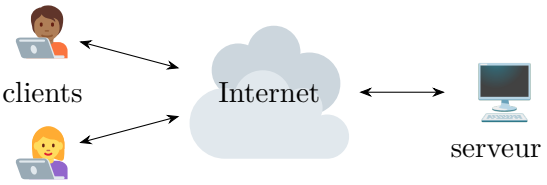
1 Le modèle client-serveur

Dans une architecture d'interaction client-serveur, le serveur est une machine qui possède des ressources et qui envoie ces ressources aux clients à la suite de requêtes de ces derniers.



Un même serveur peut répondre aux requêtes de plusieurs clients.
Et cette communication se fait bien souvent à travers le réseau Internet (par exemple lorsqu'on interroge un serveur Web) :



On peut citer par exemple des serveurs de pages Web, de mail, de fichiers...

2 Exemple du serveur Web : protocole HTTP

2.1 Présentation

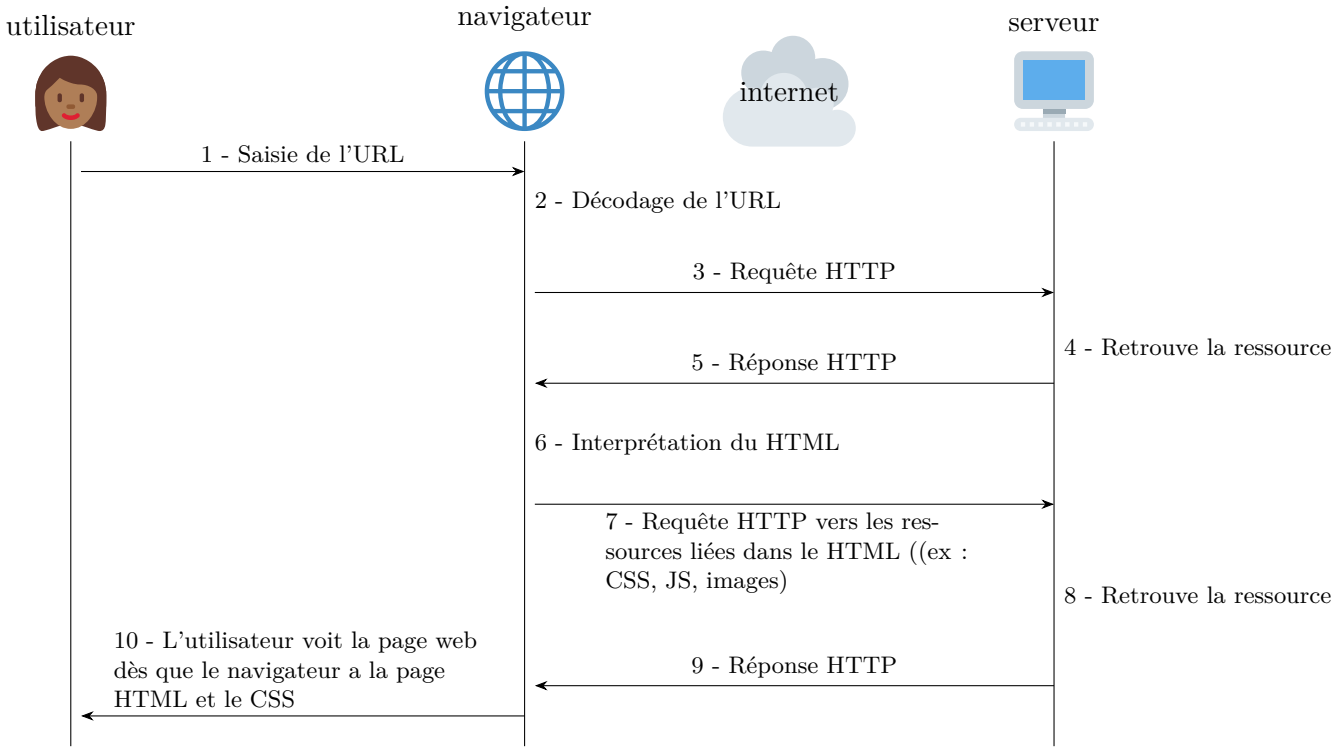
Le protocole **HTTP** (HyperText Transfert Protocol) est un protocole de la couche Application (voir chap. réseaux P5-2) mis en œuvre lorsqu'une requête est effectuée à la demande d'un logiciel navigateur (Edge, Firefox, Safari...).

L'utilisateur saisie une URL dans le navigateur, et ce dernier va engager une série de requêtes HTTP avec le serveur pour obtenir les ressources nécessaires à l'affichage complet de la page Web.
Lorsque le navigateur reçoit une page HTML, il l'interprète pour l'afficher et demande à récupérer toutes les ressources liées à la page (feuilles de style CSS, images, vidéos, scripts JavaScript). Pour cela, il va envoyer plusieurs requêtes HTTP au serveur et mettre à jour l'affichage.
Le navigateur réalise plusieurs actions en parallèle. Il procède à un premier affichage dès qu'il reçoit la page HTML et les feuilles de styles CSS liées. En parallèle, il envoie les requêtes pour récupérer les autres éléments liées à la page et fait des mises à jours de l'affichage lorsqu'il les reçoit.
La figure page suivante illustre les interactions réalisées pour afficher une page HTML. On voit que le navigateur effectue plusieurs requêtes HTTP.

Complément URL : Une URL (Uniforme Ressource Location) désigne « un point d'accès » à une ressource à travers le réseaux Internet. Elle est composée de 3 parties :

- le **protocole** de communication (ex : http, ftp).
- le **nom de domaine** d'un serveur (traduit en **adresse IP** par les serveurs DNS) . Il est parfois accompagné explicitement du numéro de port (ex : port 80 par défaut pour les serveurs Web).
- le **chemin d'accès** à la ressource dans l'arborescence du serveur.

Exemple : `https://developer.mozilla.org/fr/docs/Web/HTTP/Overview`
- protocole `https`,
- nom de domaine `developer.mozilla.org` (IP `52.84.174.32`),
- chemin vers la ressource `fr/docs/Web/HTTP/Overview`.



2.2 Structure d'une requête HTTP

On rappelle qu'un protocole définit des règles de communication. Les requêtes HTTP et leurs réponses respectent donc des règles de mise en forme.
Une requête HTTP présente le format suivant :

```
Ligne de commande (Commande, URL, Version de protocole)
En-tête de requête
[Ligne vide]
Corps de requête
```

Exemple de requête (avec un corps vide, fréquent) :

```
GET developer.mozilla.org/fr/docs/Web/HTTP/Overview HTTP/1.1
User-Agent : Mozilla/5.0
Accept : text/html
```

Remarque : il existe plusieurs champs facultatifs dans l'en-tête de la requête (et ceci dépend aussi de la version du protocole).

Il est possible de transmettre des informations à un site à travers l'en-tête de la requête ou dans le corps de la requête (voir chap. formulaires P4-4).

Les scripts Javascript en revanche sont exécutés au niveau du client, sans interaction avec le serveur.

2.3 Structure d'une réponse HTTP

Les réponses HTTP présentent le format suivant :

```
Ligne de statut (Version, Code-réponse, Texte-réponse)
En-tête de réponse
[Ligne vide]
Corps de réponse
```

Exemple de Réponse :

```
HTTP/1.1 200 OK
Date: Thu, 15 feb 2023 12:02:32 GMT
Server: Apache/2.0.54 (Debian GNU/Linux) DAV/2 SVN/1.1.4
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=ISO-8859-1

<!DOCTYPE html>
<html lang="fr">
<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>
```

Dans cet exemple, le code réponse `200 (OK)` indique que le document recherché par le client a bien été trouvé par le serveur. La page Web va alors s'afficher correctement sur le client.

On pourra retenir 2 autres codes de réponses assez classiques :

- `404` : page non trouvée (c'est souvent parce que l'URL a été mal écrite dans la barre d'adresse du navigateur, ou parce que le fichier a été déplacé ou retiré sur le serveur).
- `403` : accès refusé : le serveur a compris la requête mais les droits d'accès ne permettent pas au client d'accéder à la ressource.

Remarque : les navigateurs proposent des « outils de développement » (raccourci touche F12 en général) permettant entre autres d'obtenir des informations sur les requêtes échangées.

3 Protocole HTTPS

Le protocole HTTP ne chiffre pas les données transmises sur le réseau. Si un « pirate » intercepte les paquets échangés sur le réseau, il pourra donc connaître l'intégralité de la communication. Ceci est évidemment à éviter dans le cas de transactions qui nécessitent du secret (*ex* : communication avec une banque...).

> Le protocole HTTPS (S pour Secure) permet de chiffrer les données avant leur transmission.
> Ce protocole assure donc une confidentialité des échanges sur le réseau.