

Interactions entre l'homme et la machine sur le Web

J. Boucher

Lycée Pierre-Paul RIQUET, Première NSI

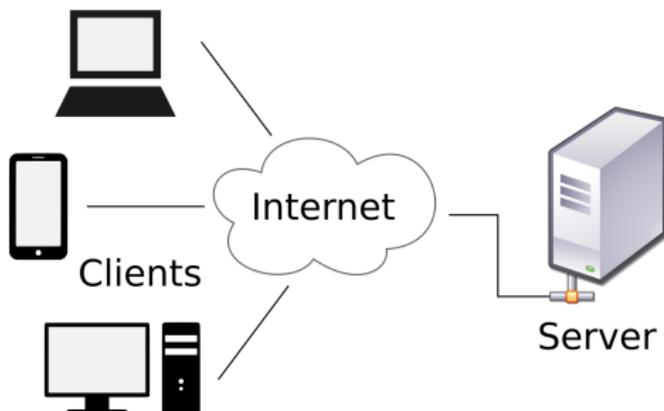
11 janvier 2025

Plan

- 1 II. Interaction client-serveur
 - 1. Modèle client-serveur
 - 2. Protocole HTTP
 - 3. Passage de paramètres

1. Modèle client-serveur

- La communication entre deux machines connectés à internet s'effectue le plus souvent selon un **modèle client-serveur**.



- C'est le aussi cas du Web, qui utilise ce modèle pour faire interagir un **navigateur Web** (le client) avec un **serveur Web**.

2. Protocole HTTP

- Le **protocole HTTP** (HyperText Transfer Protocol) ou littéralement « protocole de transfert hypertexte » est un protocole de type client-serveur qui définit les messages envoyés entre le navigateur (client) appelé et le serveur Web (serveur).



- Les messages envoyés par le client sont appelés des **requêtes**, ceux envoyés en retour par le serveur sont appelés des **réponses**.

Schéma de l'interaction **client**-serveur

Lorsqu'un utilisateur utilise son navigateur pour accéder à la page Web `http://snt-nsi.info/1nsi/`,

- 1 le navigateur Web isole le nom du serveur `snt-nsi.info` ;
- 2 le navigateur Web effectue une requête DNS pour obtenir l'adresse IP du serveur (IPv4 : 51.178.82.21) ;
- 3 le navigateur Web se connecte à la machine dont l'adresse IP est 51.178.82.21, en utilisant le *protocole TCP* sur le port 80 ;
- 4 une fois la connexion établie, le **navigateur Web** peut effectuer la **requête HTTP** pour demander la ressource souhaitée (ici le document `/1nsi`).
- 5 le **serveur Web** renvoie une **réponse HTTP** au navigateur contenant le code HTML ;
- 6 le navigateur effectue le rendu de la page correspondante qui est affichée à l'écran.

Requête HTTP

Syntaxe d'une requête HTTP (client vers serveur)

- 1 la première ligne s'appelle la **ligne de commande**, elle contient la méthode utilisée, l'URL de la ressource demandée et la version du protocole utilisée ;
- 2 les lignes suivantes correspondent à toutes les **en-têtes de la requête** où l'on trouve : le nom du serveur, le client utilisé, le type de document demandé, etc.
- 3 une ligne vide pour séparer les en-têtes du corps de la requête (si le corps de la requête n'est pas vide)
- 4 **le corps de la requête**

Exemple de requête HTTP

En naviguant vers `http://snt-nsi.info/1nsi/`, le navigateur envoie la requête HTTP suivante :

```
GET /1nsi/ HTTP/1.1
```

```
Host: snt-nsi.info
```

```
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:133.0)
```

```
Gecko/20100101 Firefox/133.0
```

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

```
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
```

```
Accept-Encoding: gzip, deflate
```

```
Referer: http://snt-nsi.info/
```

```
DNT: 1
```

```
Sec-GPC: 1
```

```
Connection: keep-alive
```

```
Upgrade-Insecure-Requests: 1
```

```
Priority: u=0, i
```

```
Pragma: no-cache
```

```
Cache-Control: no-cache
```

```
le corps de la requête est vide
```

Les méthodes de requête HTTP

Il existe plusieurs **méthodes** pour une requête HTTP. Les plus importantes sont :

- **GET** : c'est la plus courante pour demander une ressource au serveur. Cette requête ne modifie pas la ressource.
- **HEAD** : cette méthode ne demande que des informations sur la ressource, sans demander la ressource elle-même.
- **POST** : Cette méthode est utilisée pour soumettre des données en vue d'un traitement côté serveur. C'est celle employée lorsque l'on envoie au serveur les données d'un formulaire.

Réponse HTTP

Syntaxe d'une réponse HTTP (serveur vers client)

- 1 la première ligne s'appelle la **ligne de statut**, elle indique la version du protocole utilisée, le code réponse ainsi que le texte réponse. ;
- 2 les lignes suivantes correspondent à toutes les **en-têtes de la réponse** où l'on trouve : le type de document envoyé, le type de serveur, les cookies éventuels, etc.
- 3 une ligne vide pour séparer les en-têtes du corps de la réponse (si le corps de la réponse n'est pas vide)
- 4 le **corps de la réponse** qui est le contenu de la ressource demandée

Exemple de réponse HTTP

En naviguant vers `http://snt-nsi.info/1nsi/`, le navigateur envoie la requête HTTP suivante :

```
HTTP/1.1 200 OK
```

```
Server: nginx/1.24.0 (Ubuntu)
```

```
Date: Wed, 08 Jan 2025 08:37:15 GMT
```

```
Content-Type: text/html
```

```
Last-Modified: Fri, 27 Dec 2024 15:16:50 GMT
```

```
Transfer-Encoding: chunked
```

```
Connection: keep-alive
```

```
ETag: W/"676ec4e2-61d9"
```

```
Content-Encoding: gzip
```

```
<!doctype html>
```

```
<html lang="fr" class="no-js">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
...
```

```
<title>1NSI - Les Sciences Informatiques au Lycée</title>
```

```
<link rel="stylesheet"
```

```
href="../assets/stylesheets/main.0253249f.min.css">
```

Les différents codes de réponse

On peut trouver tous les codes de réponse possibles à l'adresse :

<https://developer.mozilla.org/fr/docs/Web/HTTP/Status>.

Parmi les plus importants, on trouve :

- **200** : Lorsque la ressource est disponible, la requête est donc un succès. Le texte-réponse associé est « **OK** ».
- **404** : Lorsque la ressource est indisponible. Le texte-réponse associé est « **NOT FOUND** ».
- **403** : Lorsque la permission d'accéder à la ressource est refusée. Le texte-réponse associé est « **FORBIDDEN** ».
- **500** : Lorsque le serveur rencontre une erreur interne. Le texte réponse associé est « **INTERNAL ERROR** ».

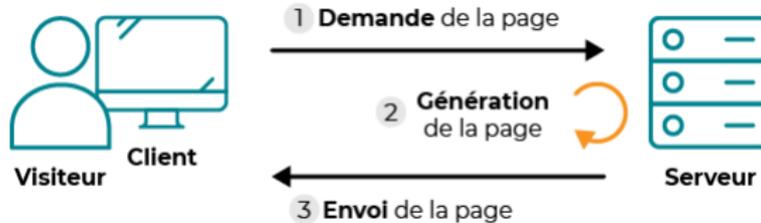
Sécurisation des échanges

- Le protocole HTTP a un problème majeur : les données sont échangées « en clair ».
→ *Toutes les machines du réseau relayant le message ont connaissance de ces données.*
- Pour protéger les données sensibles (coordonnées bancaires, mots de passe...), client et serveur utilise un protocole dit sécurisé, le **protocole HTTPS** (S pour « Secure »), permettant de **chiffrer les données**.

`https://pierre-paul-riquet.mon-ent-occitanie.fr/`

3. Passage de paramètres

- Un serveur Web peut générer le code HTML d'une page « à la demande », à partir d'informations fournies par le navigateur : on parle de page Web **dynamique**.



- Le client peut passer des paramètres au serveur lors de la requête HTTP :
 - soit directement via l'URL ;
 - soit en utilisant des éléments HTML comme les formulaires, permettant à l'utilisateur de saisir des valeurs et de les envoyer au serveur.

Attention : Il ne faut pas confondre le rendu dynamique d'une page Web par le navigateur et la génération dynamique d'une page Web par le serveur !

- Le client peut passer des **paramètres à un serveur via l'URL** de la ressource demandée en utilisant un point d'interrogation ? suivi des différents paramètres écrits sous la forme `nom=valeur`.
- S'il y a plusieurs paramètres, ceux-ci sont séparés par une esperluette &.

Syntaxe :

```
protocole://nom-ou-adresse:port/chemin?n1=v1&n2=v2& ... &nk=vk
```

où **n1**, **n2**, ..., **nk** sont les noms des paramètres et **v1**, **v2**, ..., **vk** leurs valeurs respectives.

Par exemple :

```
https://pierre-paul-riquet.mon-ent-occitanie.fr/sg.do?PROC=PAGE_ACCUEIL&ACTION=VALIDER
```