NSI · PREMIÈRE / COURS

Le Web : client, serveur et HTTP

Astrolab Cours · astrolabcours.fr

01 Le problème

Quand tu tapes une adresse et appuies sur Entrée, une page apparaît en moins d'une seconde, avec du texte, des images, un formulaire. Qui fabrique cette page, et comment arrive-t-elle jusqu'à ton écran ?

01 L'idée : le client et le serveur du restaurant

Au restaurant, le client passe commande, le serveur va en cuisine et rapporte le plat. Sur le Web, ton navigateur est le **client**, l'ordinateur qui héberge le site est le **serveur**. Le client demande, le serveur répond. La langue qu'ils parlent, c'est **HTTP**.

01 Requête et réponse

1. Le client envoie une **requête HTTP** : une méthode (GET, POST), une ressource (`/index.html`), des en-têtes.
2. Le serveur renvoie une **réponse** : un **code de statut**, des en-têtes, et le contenu (HTML, image…).

Code	Sens
200	OK
301	page déplacée
404	page introuvable
500	erreur du serveur

Une **URL** comme `https://exemple.fr/cours/index.html?niveau=1re` donne le protocole, le nom du serveur, le chemin de la ressource et des paramètres.

La page HTML

Le serveur envoie du **HTML**, qui décrit la structure : titres, paragraphes, liens, images. Le **CSS** gère l'apparence, le **JavaScript** l'interactivité.

```
<form action="/inscription" method="post">
  <input type="text" name="pseudo">
  <button type="submit">Envoyer</button>
</form>
```

© Astrolab Cours · astrolabcours.fr/nsi/premiere/web-et-http/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

GET et POST

- **GET** : les paramètres sont ajoutés à l'URL (`?pseudo=lea`). Pratique pour une recherche, qu'on peut mettre en favori.
- **POST** : les données voyagent dans le corps de la requête, invisibles dans l'URL. Pour un mot de passe ou un formulaire qui modifie des données, on utilise POST.

Côté client, côté serveur

Un script JavaScript s'exécute **dans le navigateur** (côté client) : il réagit à un clic sans recharger la page. Un programme PHP ou Python s'exécute **sur le serveur** : il lit la base de données et fabrique la page. Le client ne voit jamais le code du serveur.

HTTPS

HTTPS, c'est HTTP chiffré : un espion sur le réseau voit qu'un échange a lieu, mais pas son contenu.

Les pièges

- POST n'est pas chiffré pour autant : seul HTTPS chiffre.
- Le JavaScript de la page est visible par tous ; les vérifications de sécurité doivent aussi être faites sur le serveur.
- 404 est une erreur du client (mauvaise adresse), 500 une erreur du serveur.

© Astrolab Cours · astrolabcours.fr/nsi/premiere/web-et-http/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits