IHM Web et protocole HTTP

Fiche de synthèse

1 Interface homme-machine

Une **interface homme-machine** (IHM) est tout ce qui permet à une personne de communiquer avec une machine : matériel et logiciel qui réagit aux gestes. Le navigateur Web en est l'exemple quotidien.

2 Le protocole HTTP

Définition 1 **(HTTP)** Un **protocole** est un ensemble de règles que deux machines suivent pour se comprendre. **HTTP** fixe la façon dont un **client**, le navigateur, et un **serveur Web**, comme Apache ou nginx, échangent des messages : une **requête** du client, puis une **réponse** du serveur.

Afficher http://www.exemple.fr/cours/http.html.

1. Le navigateur isole le nom de domaine `www.exemple.fr`.
2. Un serveur DNS lui donne l'adresse IP ; il ouvre une connexion TCP.
3. Il envoie une requête HTTP qui demande `/cours/http.html`.
4. Le serveur renvoie le fichier dans une réponse HTTP.
5. Le navigateur lit le code HTML reçu et construit la page.

3 La requête

```
GET /cours/http.html HTTP/1.1
Host: www.exemple.fr
User-Agent: Mozilla/5.0 ... Firefox/128.0
Accept: text/html
```

Première ligne : **méthode**, chemin, version. Puis des **champs** `Nom: valeur` : `Host` (domaine visé), `User-Agent` (navigateur), `Referer` (page d'où l'on vient).

Méthodes : `GET` demande une ressource (clic sur un lien), `POST` envoie des données à traiter, `PUT` dépose ou remplace une ressource, `DELETE` la supprime.

4 La réponse

```
HTTP/1.1 200 OK
Date: Tue, 13 Oct 2026 08:15:42 GMT
Server: nginx/1.26.2
Content-Type: text/html; charset=UTF-8
```

La **ligne de statut** donne un **code de réponse**. Après l'entête et une ligne vide vient le **corps**, le code HTML que le navigateur interprète. `Content-Type` donne le type de la ressource, `Location` l'adresse d'une redirection.

Code	Signification
200	la requête a réussi
301, 302	ressource déplacée, définitivement ou provisoirement
400	requête mal formée
401, 403	identification requise ; accès refusé
404	aucune ressource à cette adresse
500	erreur du serveur

Le premier chiffre donne la famille : 2 succès, 3 redirection, 4 erreur du client, 5 erreur du serveur.

5 Formulaires : GET et POST

La balise `<form>` envoie les champs saisis au programme désigné par son attribut `action`, avec la méthode de son attribut `method`. Chaque champ est transmis sous la forme `name=valeur`.

— `GET` : paramètres ajoutés à l'adresse, visibles (`/inscription?prenom=Leo&age=16`), pour une recherche.

— `POST` : paramètres dans le corps de la requête ; pour un envoi qui modifie des données. Pas chiffré pour autant : seul HTTPS protège.

Ordre : script JavaScript éventuel sur le **client**, envoi de la requête, programme du **serveur** qui fabrique la page, affichage. Un **cookie**, reçu par `Set-Cookie`, est mémorisé par le client et renvoyé au site à chaque requête (champ `Cookie`).

6 HTTPS

HTTP circule en clair. **HTTPS** chiffre les échanges : le serveur présente un **certificat**, délivré par un organisme de confiance et vérifié par le navigateur, puis client et serveur conviennent d'une clé de chiffrement. Le certificat protège contre l'attaque de l'intermédiaire ; le chiffrement rend illisibles les données interceptées. Avant de saisir un mot de passe, on vérifie que l'adresse commence par `https://`.

À retenir.

— HTTP : une requête du client, une réponse du serveur, en texte lisible.

— `GET` met les paramètres dans l'adresse, `POST` dans le corps de la requête.

— Le code 200 signale un succès, 404 une ressource introuvable.