

# Introduction aux Protocoles Web et à HTTP(S)

## 1. Différences entre HTTP et HTTPS

- **HTTP :**

*Définition :* (HyperText Transfer Protocol) est un protocole de communication utilisé pour transférer des données sur le web. Les données sont envoyées en **texte clair**, ce qui peut être intercepté par des tiers malveillants.

- *Caractéristiques :* Protocole non sécurisé

- **HTTPS :**

*Définition :* (**HTTP Secure**) est la version sécurisée de HTTP. Il utilise un chiffrement via TLS/SSL pour protéger les données échangées entre le client et le serveur. Cela garantit confidentialité, intégrité des données et authentification du serveur.

- *Caractéristiques :*

- protocole sécurisé en vérifiant l'identité du site web(via un certificat SSL ou TLS)
- Protège contre l'espionnage (attaque de type man in middle, usurpation de sites...
- son utilisation est fortement recommandée pour protéger les données sur le web

- **Résumé des différences :**

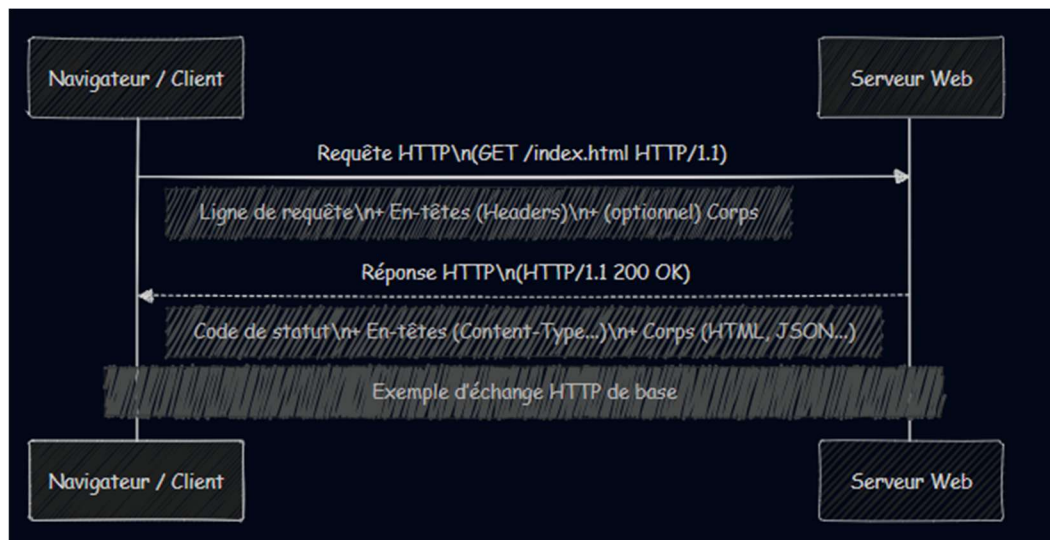
- HTTP = communication non chiffrée donc non sécurisée
- HTTPS = communication sécurisée par chiffrement

---

## 2. Structure d'une requête HTTP et d'une réponse HTTP

### Requête HTTP (exemple : GET)

Un requête HTTP permet de demander une ressource (site web, image , données d'API) à un serveur WEB. C'est le principal moyen de communication entre un client et un serveur via un protocole HTTP .



La requête comporte trois parties principales : La ligne de requête, le header et le corps.

Le client fait une requête http avec une méthode (par exemple ici GET, pour demander une information). Cette méthode envoie une question au serveur Web.

Le serveur Web renvoie une réponse au Client avec un code de status ainsi que les informations qui ont été demandées par le client.

Une requête se compose **d'une ligne de requête** avec : une méthode http(GET, POST etc), d'un chemin ou d'une ressource(URL par exemple) et de la version du protocole.

Elle est composée **d'un en-tête**(header) qui possèdent d'autres informations à envoyer au serveur :

Host(nom du site), User-Agent(navigateur, outil utilisé), Accept(type de contenu autorisé)

Enfin un corps (body) qui est optionnel qui contient des données à envoyer (JSON, fichier).

### Réponse HTTP (exemple) :

Une réponse HTTP est composée de trois parties également :

- La ligne de statut qui indique le résultat de la requête avec un code statut
- Les en-têtes(header) qui sont des lignes d'informations qui donnent le détail de la réponse (type de contenu, taille, règles de cache, cookies à stocker)
- Le corps(body) qui est le contenu réel renvoyé par le serveur (page HTML, données JSON, PDF etc)

---

### 3. Méthodes HTTP courantes

Les méthodes HTTP les plus courantes pour les requêtes sont les suivantes :

Méthode	Description	Cas d'usage typique
GET	Pour récupérer des données	Charger une page web, récupérer des données d'une API
POST	Envoyer des données	Soumettre un formulaire, créer une ressource sur un serveur
PUT	Mettre à jour une ressource	Modifier complètement une ressource existante
PATCH	Modifier partiellement une ressource	Mettre à jour certains champs d'une ressource
DELETE	Supprimer une ressource	Supprimer un élément sur un serveur
HEAD	Récupérer uniquement les headers	Vérifier si une ressource existe sans télécharger son contenu

---

### 4. Codes de statut HTTP

Les codes statuts ont tous une signification.

- **1XX Information** (la requête est reçue et le processus continue)
- **2XX Success** (la requête a réussi)
- **3XX Redirection** (Le client doit effectuer une action supplémentaire pour terminer la requête)
- **4XX Erreur côté Client** (La requête contient une erreur, souvent causée par le client)
- **5XX Erreur côté Serveur** (Le serveur a échoué à traiter une requête valide.)

Voici les codes statut les plus courants :

Code	Description	Scénario d'utilisation typique
200	OK	Requête réussie, ressource retournée
301	Moved Permanently	Redirection permanente vers une autre URL
400	Bad Request	Requête mal formée ou invalide

<b>Code</b>	<b>Description</b>	<b>Scénario d'utilisation typique</b>
401	Unauthorized	Accès refusé, authentification requise
403	Forbidden	Accès refusé malgré authentification
404	Not Found	Ressource demandée inexistante sur le serveur
500	Internal Server Error	Erreur interne du serveur

---