

Anatomie d'une requête HTTP

HTTP (*HyperText Transfer Protocol*) est le protocole du web. Le principe est simple : un **client** envoie une **requête**, le **serveur** renvoie une **réponse**. Toujours dans ce sens, toujours une réponse par requête.

Voir HTTP en vrai

L'option `-v` (verbose) de `curl` affiche tout l'échange :

```
curl -v https://formation.tfrichet.fr
```

Les lignes qui commencent par `>` sont envoyées par le client, celles qui commencent par `<` sont reçues du serveur.

La requête

Exemple avec [JSONPlaceholder](#), une fausse API publique faite pour s'entraîner :

```
curl -v https://jsonplaceholder.typicode.com/posts \  
-H "Content-Type: application/json" \  
-d '{"title": "Hello", "body": "Mon premier article", "userId": 1}'
```

```
POST /posts HTTP/1.1  
Host: jsonplaceholder.typicode.com  
User-Agent: curl/8.10.1  
Accept: */*  
Content-Type: application/json  
  
{"title": "Hello", "body": "Mon premier article", "userId": 1}
```

- **Ligne de requête** : la méthode (`POST`), le chemin (`/posts`) et la version de HTTP.
- **Headers** : des paires `Nom: valeur` qui donnent des informations sur la requête.

- **Ligne vide**, puis le **body** (facultatif) : les données envoyées.

La réponse

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8
Location: https://jsonplaceholder.typicode.com/posts/101

{"title": "Hello", "body": "Mon premier article", "userId": 1, "id": 101}
```

- **Ligne de statut** : la version, le **code de statut** et sa description.
- **Headers**, ligne vide, puis le **body**.

Les headers à connaître

Header	Sens	Rôle
Host	Requête	Le nom de domaine visé (plusieurs sites peuvent partager un serveur)
Content-Type	Les deux	Le format du body envoyé : application/json, text/html...
Accept	Requête	Le format que le client souhaite recevoir
Authorization	Requête	Les informations d'authentification
User-Agent	Requête	Le logiciel client
Set-Cookie / Cookie	Réponse / Requête	Le serveur dépose un cookie, le client le renvoie ensuite
Location	Réponse	L'adresse d'une ressource créée ou d'une redirection

Oublier `Content-Type: application/json` en envoyant du JSON est l'erreur numéro 1. Le serveur ne sait pas comment lire le body et répond souvent par une erreur peu claire.

Un protocole sans état

HTTP est **sans état** (*stateless*) : le serveur ne se souvient de rien entre deux requêtes. Chaque requête doit contenir tout ce qu'il faut pour être comprise.

Alors comment un site sait-il que vous êtes connecté ? Le client renvoie à chaque requête une preuve d'identité : un **cookie** pour un site web, un **token** ou une **clé d'API** pour une API.

HTTP n'a pas de mémoire persistante. Ce n'est pas un défaut : c'est ce qui permet de répartir les requêtes sur plusieurs serveurs.

Les familles de codes de statut

Famille	Signification	Exemples
1xx	Information, patientez	101 Switching Protocols
2xx	Succès	200 OK, 201 Created, 204 No Content
3xx	Redirection, c'est ailleurs	301 Moved Permanently, 304 Not Modified
4xx	Erreur du client : c'est ta faute	400 Bad Request, 401, 404 Not Found
5xx	Erreur du serveur : c'est la mienne	500 Internal Server Error, 503 Service Unavailable

Le détail des codes est vu au chapitre 2.

Et HTTPS, HTTP/2, HTTP/3 ?

- **HTTPS** : le même HTTP, mais chiffré par TLS.
- **HTTP/2** et **HTTP/3** : le transport est plus rapide (multiplexage, QUIC), mais les méthodes, headers et codes restent les mêmes. Tout ce cours s'applique.

Revision #5

Created 2 October 2026 20:00:39 by Thibaud FRICHET

Updated 11 October 2026 20:02:38 by Thibaud FRICHET