

- Requête / liée à la réponse
- Fonctions / txt dans la requête

Request Parsing

- Obligatoire utile
- Obligatoire inutile
- Facultatif utile
- Facultatif inutile

GET:

Request Line: contient la méthode, la ressource demandée et la version

Host: Nom du serveur cible (utile si plusieurs domaines sur la même IP, pour distinguer plusieurs "virtual hosts", si non ignorable)

Accept: Indique les types MIME que le client accepte (text/html, application/json, etc), utile pour choisir le bon Content-Type dans la réponse

Accept-Language: Langue préférée du client, utile pour renvoyer du contenu localisé

Accept-Encoding: format de compression supportés (gzip, deflate) utile pour compresser la réponse

Connection: Gère la persistance (keep-alive ou close), si close fermer le socket après send

Authorization: données d'authentifications (Basic, Bearer, etc), utile pour gérer un accès restreint

Cookies: données de session envoyées au serveur, utile si on veut gérer des sessions

If-modified-since: permet de conditionner un GET (cache), pour renvoyer 304 Not Modified si le fichier n'a pas changé

If-None-Match: cache basé sur ETag, même usage que If-Modified-Since

Cache-Control: contrôle la mise en cache côté client, possible pour le traitement mais utile à renvoyer dans la réponse

Range: demande une partie de ressource, si on veut supporter le téléchargement partiel, **206 Partial Content**

Date: date de la requête

Upgrade-Insecure-Requests: indique préférence pour HTTPS

Referer: URL d'où vient la requête

User-Agent: identifie le client (navigateur, script, etc)

POST:

Request Line: contient la méthode, la ressource demandée et la version

Host: Nom du serveur cible (utile si plusieurs domaines sur la même IP, pour distinguer plusieurs "virtual hosts", si non ignorable)

Content-Type: type MIME du body (POST/PUT), obligatoire si body, dit comment parser le corps de la requête, (ex: JSON ou formulaire)

Content-Length: Taille (en octets) du body, obligatoire si body, pour savoir combien d'octets lire avec recv

Accept: Indique les types MIME que le client accepte (text/html, application/json, etc), utile pour choisir le bon Content-Type dans la réponse

Accept-Language: Langue préférée du client, utile pour renvoyer du contenu localisé

Accept-Encoding: format de compression supportés (gzip, deflate), utile pour compresser la réponse

Connection: Gère la persistance (keep-alive ou close), si close fermer le socket après send

Transfer-Encoding: corps envoyé en morceaux (chunked), si on veut supporter chunked

Authorization: données d'authifications (Basic, Bearer, etc), utile pour gérer un accès restreint

Cookies: données de session envoyées au serveur, utile si on veut gérer des sessions

Origin: domaine d'origine de la requête, si on veux implémenter CORS, si non ignorable

Cache-Control: contrôle la mise en cache côté client, pas utile pour le traitement mais utile à renvoyer dans la réponse

Expect: indique un comportement attendu (ex: 100-continue) → renvoyer HTTP/1.1 100 Continue avant le body

Date: date de la requête

Referer: URL d'où vient la requête

User-Agent: identifie le client (navigateur, script, etc)

X-Requested-With: nom standard (souvent AJAX)

DELETE:

Request Line: contient la méthode, la ressource demandée et la version

Host: Nom du serveur cible (utile si plusieurs domaines sur la même IP, pour distinguer plusieurs "virtual hosts", si non ignorable)

Content-Type: type MIME du body (POST/PUT), obligatoire si body, dit comment parser le corps de la requête, (ex: JSON ou formulaire)

Content-Length: Taille (en octets) du body, obligatoire si body pour savoir combien d'octets lire avec recv

Accept: Indique les types MIME que le client accepte (text/html, application/json, etc), utile pour choisir le bon Content-Type dans la réponse

Accept-Language: Langue préférée du client, utile pour renvoyer du contenu localisé

Accept-Encoding: format de compression supportés (gzip, deflate) utile pour compresser la réponse

Connection: Gère la persistance (keep-alive ou close), si close fermer le socket après send

Transfer-Encoding: corps envoyé en morceaux (chunked), si on veut supporter chunked

Authorization: données d'authentifications (Basic, Bearer, etc), utile pour gérer un accès restreint

Cookies: données de session envoyées au serveur, utile si on veut gérer des sessions

Origin: domaine d'origine de la requête, si on veux implémenter CORS, si non ignorable

Cache-Control: contrôle la mise en cache côté client, pas utile pour le traitement mais utile à renvoyer dans la réponse

Date: date de la requête

User-Agent: identifie le client (navigateur, script, etc)

X-Requested-With: nom standard (souvent AJAX)

Retour Serveur

Status Line: ex: HTTP/1.1 200 OK, indique la version, le code HTTP et le message

Date: Donne la date / heure de la réponse (GMT), avec STD::time

Content-Type: indique le MIME du contenu (text/html, text/plain, application/json), obligatoire si body, crucial pour que curl / navigateur sache lire la réponse

Content-Length: taille du body en octet, inutile si Transfer-Encoding utilisé

Connection: Indique si la connection reste ouverte (keep-alive) ou se ferme (close), permet de fermer proprement le socket

Server: identifie le serveur (ex: Server: MyServer/0.1), utile pour le débug sinon inutile

Cache-Control: indique comment le client peut mettre en cache la réponse (no-cache, max-age=...), utile pour gérer le cache

Last-modified: Date de dernière modification du contenu, utile si on implemente le cache (If-Modified-Since)

Location: utilisé pour les redirections (3xx), utile si on fait des 302 ou 301

Allow: liste les méthodes HTTP autorisée (pour 405 Method Not Allowed), pour répondre à des méthodes non supportées

Set-Cookie: envoie des cookies au client, si on gère des sessions

WWW-Authenticate: indique le schéma d'authentification demandé (`Basic realm="..."`), si on ne réussit pas à authentifier l'utilisateur, on renvoie un `401 Unauthorized`

Transfer-Encoding: ex: `chunked`, si la taille du body est inconnue

ETag: identifiant unique du contenu (pour le cache)

Access-Control-Allow-Origin: CORS: autorise certains domaines à accéder à la ressource

Content-Encoding: indique la compression utilisée (`gzip`, `deflate`)

Expires: date d'expiration de la ressource (cache)

Retry-After: délai avant une nouvelle tentative (`503` ou `429`)

Content-Range: indique la portion renvoyée d'une ressource (`206 Partial Content`)