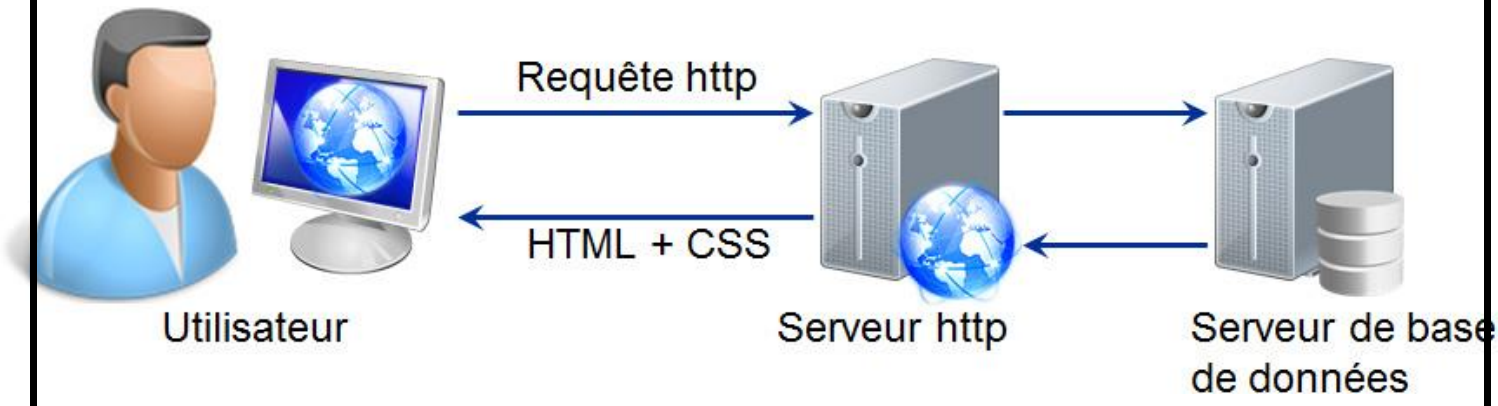
2026



MR GIBERT
CAMILLIA MASKAR-AOURAGH & MOKNI ZYED

Table des matières

Introduction

Le WWW, qui signifie World Wide Web (en français « toile » ou « réseau mondial »), désigne l'ensemble des pages et des sites web accessibles sur Internet à l'aide d'un navigateur comme Chrome, Firefox ou Safari. Par exemple, www.wikipedia.org est un site faisant partie du World Wide Web. Il est important de distinguer Internet, qui correspond au réseau mondial d'infrastructures, de câbles et de serveurs, du Web (WWW), qui est un service fonctionnant sur Internet parmi d'autres, comme le courrier électronique, le FTP ou la messagerie. Aujourd'hui, le préfixe www est souvent optionnel et relève principalement d'une habitude historique.

Une URL (*Uniform Resource Locator*, en français « localisateur uniforme de ressource ») est l'adresse exacte permettant d'accéder à une ressource sur Internet, comme une page web, une image, une vidéo ou un fichier. Par exemple, dans l'URL https://www.example.com/page.html, la partie https:// correspond au protocole de communication, www.example.com désigne le nom du serveur et /page.html représente la ressource demandée. En résumé, une URL fonctionne comme une adresse postale, car elle indique précisément où se trouve un contenu sur Internet.

Le Web est l'abréviation de World Wide Web et signifie littéralement « toile », en référence à une toile d'araignée, car les pages sont reliées entre elles par des liens hypertextes. Il permet aux utilisateurs de consulter des pages et des sites, de naviguer en cliquant sur des liens et d'accéder à différents contenus tels que du texte, des images et des vidéos grâce à Internet.

Analyse d'une trame http

Le protocole DNS & TCP/IP

Le modèle Client-Serveur est un mode de fonctionnement des réseaux dans lequel un client (ordinateur ou application) envoie une requête pour demander un service, et un serveur reçoit cette demande, la traite et renvoie une réponse. Le réseau permet l'échange entre les deux, comme dans un restaurant où le client commande un plat et la cuisine le prépare et le sert.

Le protocole DNS (Domain Name System) permet de traduire les noms de domaine lisibles par l'humain en adresses IP compréhensibles par les ordinateurs ; il utilise généralement le protocole UDP sur le port 53. Les communications réseau reposent sur la suite TCP/IP, qui fonctionne au niveau transport selon deux modes : le mode connecté avec TCP, assurant une connexion fiable entre le client et le serveur (utilisé notamment pour le Web), et le mode non connecté avec UDP, plus rapide mais sans garantie de réception. La communication entre deux applications se fait à l'aide de sockets, qui nécessitent une adresse IP et un numéro de port, chaque protocole utilisant un port spécifique (par exemple HTTP 80, DNS 53, FTP 21).

Un pare-feu a pour rôle de sécuriser une machine en bloquant certains numéros de ports ou adresses IP. Un protocole est un ensemble de règles qui définit la manière dont les données sont échangées et comprises lors d'une communication sur un réseau.

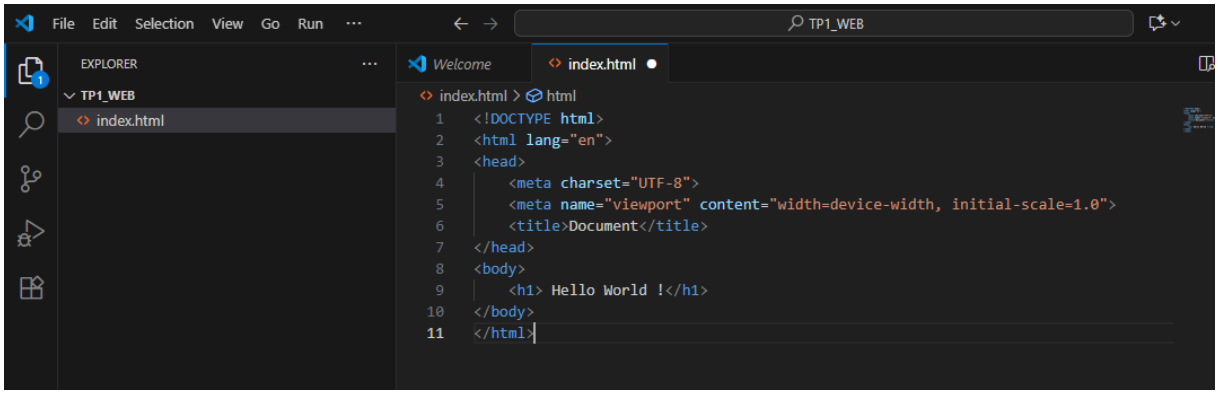
Le protocole http

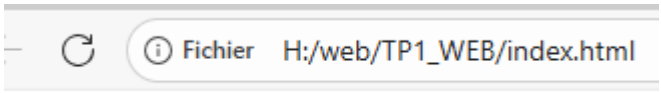
Le protocole HTTP repose sur l'architecture Client-Serveur. Le client HTTP est généralement un navigateur web (comme Chrome ou Firefox), tandis que le serveur HTTP est un logiciel serveur tel qu'Apache, Nginx ou IIS. Le client et le serveur sont donc tous deux des logiciels qui doivent respecter les règles de communication du protocole HTTP pour pouvoir dialoguer correctement. Ce fonctionnement peut être étudié à l'aide de captures de trames réseau ou en communiquant directement en utilisant le langage du protocole http

Protocole	Couche OSI
HTTP ; FTP ; Telnet ; DNS	Couche 7 : Appliccation
TCP ; UDP	Couche 4 : Transport
IP	Couche 3 : Réseau
Ethernet	Couche 2 : Liaison de données

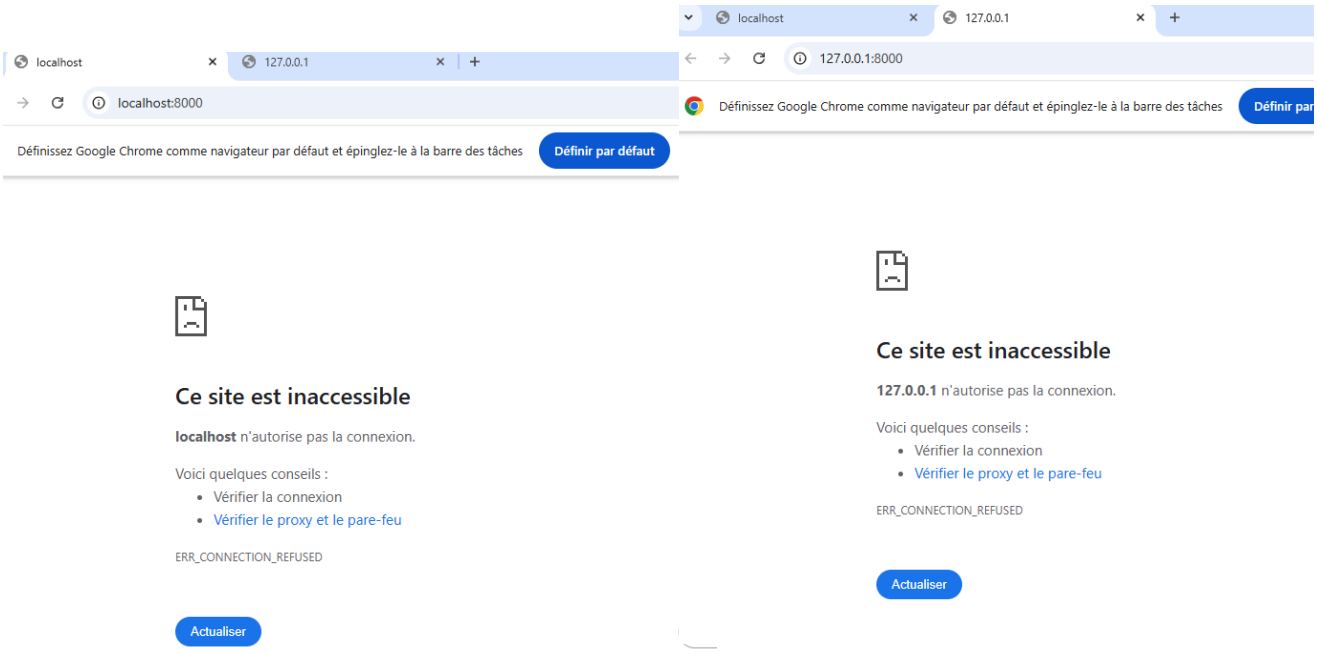
Les différents modes

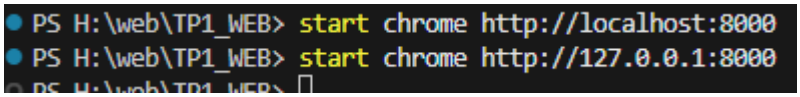
Mode fichier Local





Mode fichier Localhost





Etant donné la présence du pare-feu sur le pc nous n'avons pas accès au résultat mais étant indiqué sur le sujet nous comprenons qu'il y a une partie interface client et une autre partie Serveur dans une même machine.

Mode client-server

```
PS H:\web\TP1_WEB> netstat -a -n
Connexions actives

  Proto  Adresse locale         Adresse distante       État
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:623            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5040           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5357           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5432           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:7680           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:16992          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49669          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49702          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:49671        127.0.0.1:49672        ESTABLISHED
  TCP    127.0.0.1:49672        127.0.0.1:49671        ESTABLISHED
  TCP    172.31.1.84:49999      74.242.255.116:443     ESTABLISHED
  TCP    172.31.1.84:51192      173.194.76.188:5228    ESTABLISHED
```

```
  TCP    [::]:135               [::]:0                 LISTENING
  TCP    [::]:445               [::]:0                 LISTENING
  TCP    [::]:623               [::]:0                 LISTENING
  TCP    [::]:5357              [::]:0                 LISTENING
  TCP    [::]:5432              [::]:0                 LISTENING
  TCP    [::]:7680              [::]:0                 LISTENING
  TCP    [::]:16992             [::]:0                 LISTENING
  TCP    [::]:49664             [::]:0                 LISTENING
  TCP    [::]:49665             [::]:0                 LISTENING
  TCP    [::]:49666             [::]:0                 LISTENING
  TCP    [::]:49667             [::]:0                 LISTENING
  TCP    [::]:49668             [::]:0                 LISTENING
  TCP    [::]:49669             [::]:0                 LISTENING
  TCP    [::]:49702             [::]:0                 LISTENING
  TCP    [::1]:42050            [::]:0                 LISTENING
  TCP    [::1]:49670            [::]:0                 LISTENING
  UDP    0.0.0.0:123            *:*
```

```
  TCP    172.31.1.84:54666      52.97.201.18:443       ESTABLISHED
  TCP    172.31.1.84:54880      98.66.133.185:443      ESTABLISHED
  TCP    172.31.1.84:54886      172.31.1.9:445         ESTABLISHED
  TCP    172.31.1.84:54903      98.66.133.185:443      ESTABLISHED
  TCP    172.31.1.84:55091      23.53.1.33:443         ESTABLISHED
  TCP    172.31.1.84:55989      216.58.215.35:443      ESTABLISHED
  TCP    172.31.1.84:56381      142.250.179.65:443     TIME_WAIT
  TCP    172.31.1.84:56564      216.58.215.35:443      ESTABLISHED
  TCP    172.31.1.84:57618      13.107.213.61:443      ESTABLISHED
  TCP    172.31.1.84:57898      52.98.227.130:443      ESTABLISHED
  TCP    172.31.1.84:58336      142.250.75.234:443     ESTABLISHED
  TCP    172.31.1.84:58496      142.250.178.133:443    ESTABLISHED
  TCP    172.31.1.84:58782      142.251.142.10:443     TIME_WAIT
  TCP    172.31.1.84:59034      23.53.1.39:443         ESTABLISHED
  TCP    172.31.1.84:59383      216.58.215.35:443      ESTABLISHED
  TCP    172.31.1.84:60215      23.48.32.241:443       CLOSE_WAIT
  TCP    172.31.1.84:60216      23.203.61.38:443       ESTABLISHED
  TCP    172.31.1.84:60219      23.53.1.42:443         ESTABLISHED
  TCP    172.31.1.84:60220      20.42.73.26:443        ESTABLISHED
  TCP    172.31.1.84:61694      142.250.74.234:443     TIME_WAIT
  TCP    172.31.1.84:61948      216.58.215.35:443      ESTABLISHED
  TCP    172.31.1.84:62271      108.177.15.94:443      TIME_WAIT
  TCP    172.31.1.84:62555      23.203.61.41:443       ESTABLISHED
  TCP    172.31.1.84:63014      52.111.231.66:443      ESTABLISHED
  TCP    172.31.1.84:63016      52.108.240.61:443      ESTABLISHED
  TCP    172.31.1.84:63017      52.111.231.66:443      ESTABLISHED
  TCP    172.31.1.84:64218      23.50.143.207:443      CLOSE_WAIT
  TCP    172.31.1.84:65198      142.251.173.95:443     ESTABLISHED
  TCP    172.31.1.84:65199      13.107.138.10:443      ESTABLISHED
  TCP    172.31.1.84:65212      23.203.61.38:443       ESTABLISHED
  TCP    172.31.1.84:65374      142.250.74.238:443     ESTABLISHED
```

Etude du protocole http

La Requête

```
PS H:\web\TP1_WEB>  python  -m http.server 8000
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
```

Etant donné que nous ne sommes pas administrateur sur la machine nous avons décidé de faire la requête http sur la même machine

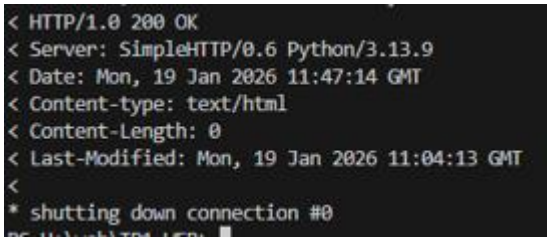
```
PS H:\web\TP1_WEB> curl -v http://127.0.0.1:8000
*   Trying 127.0.0.1:8000...
* Established connection to 127.0.0.1 (127.0.0.1 port 8000) from 127.0.0.1 port 61973
* using HTTP/1.x
> GET / HTTP/1.1
> Host: 127.0.0.1:8000
> User-Agent: curl/8.16.0
> Accept: */*
>
* Request completely sent off
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: SimpleHTTP/0.6 Python/3.13.9
< Date: Mon, 19 Jan 2026 11:47:14 GMT
< Content-type: text/html
< Content-Length: 0
< Last-Modified: Mon, 19 Jan 2026 11:04:13 GMT
<
* shutting down connection #0
PS H:\web\TP1_WEB>
```

Méthodes HTTP	
GET	Elle permet de demander une ressource au serveur
POST	Elle permet d'envoyer des données au serveur pour créer une ressource
PUT	Elle permet de remplacer complètement une ressource qui existe déjà
PATCH	Elle permet de modifier une ressource partiellement
DELETE	Elle permet de supprimer une ressource
HEAD	Elle permet d'obtenir uniquement les entêtes de la réponse sans le contenue
OPTIONS	Elle permet de connaître les méthodes autorisées par le serveur
TRACE	Elle permet de tester le chemin parcouru par la requête jusqu'au serveur
CONNECT	Elle permet de rétablie un tunnel vers le serveur.

HyperText Transfer Protocole (http) est un protocole qui permet au client (la partie donc navigateur) et au serveur web d'échanger des ressources comme des pages HTML par exemple.

La méthode GET est utilisé d'après nos résultats car elle demande au serveur la page située à la racine du site, on sait également que c'est la version http/1.1

La Réponse

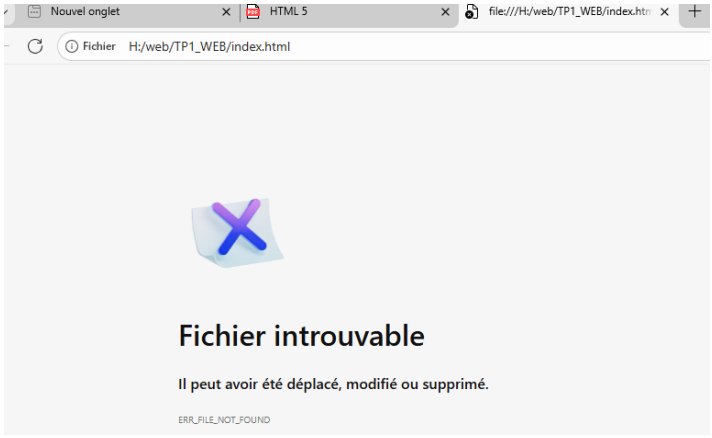


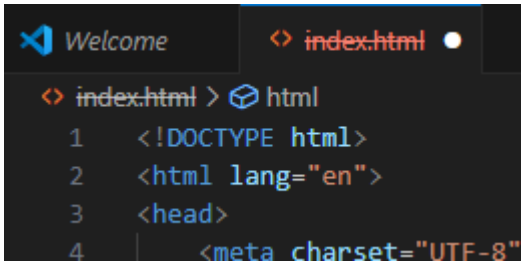
La réponse http est composée d'une ligne de statut, des en-têtes et d'un corps. Dans notre test, le serveur a répondu avec le code `< HTTP/1.0 200 OK`, ce qui signifie que la requête a été correctement reçue et traitée.

Les en-têtes indiquent les informations sur le serveur, la date et le type de contenue. Le corps correspond au code HTML de la page. Une autre requête a renvoyé le code 404 Not Found que si la ressource demandée n'existe pas. Cette erreur est sûrement due à un mauvais nom du fichier ou encore qui n'est pas présent, cette erreur peut être vite « réparé » en vérifiant le chemin ou encore le nom du fichier.
Le statu 200 est présent car le fichier est bien présent et le serveur a pu donc bien l'envoyer. Le corp correspond ici au code HTML de la page, étant donné que notre fichier est très court.

Etude des fonctionnalités d'un serveur web

Un serveur web sert à héberger et diffuser des pages web sur internet ou sur un réseau local. Il reçoit les requêtes http envoyées par les clients et renvoie les ressources demandées comme les pages HTML ou encore des fichiers CSS ou JavaScript. Un serveur web permet de traiter les requêtes http, envoyer des pages web, héberger des sites, gérer les erreurs (comme 404), assurer la sécurité des échanges, il peut aussi gérer les connexions en même temps. Les options proposées par un serveur web sont le choix du port d'écoute ; la configuration des dossiers racines appelée DocumentRoot ; la gestion des droits d'accès ; la redirection des URL ; les journaux de connexions aussi appelées les logs ; la compression de données ou encore la gestion de plusieurs sites sur un mêmes serveur.

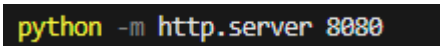




Lorsqu'on renomme le fichier index.html en index1.html, la page ne s'affiche plus automatiquement dans le navigateur. A la place le serveur affiche une erreur. Le serveur va chercher automatiquement un fichier nommé index.html. Cette erreur est présente parce que index.html c'est le nom standard reconnue par les serveurs web comme page d'accueil d'un site. S'il n'existe pas alors le serveur affichera une erreur car il ne sait pas quelle page faut afficher par défaut.

Pour afficher la page il faut taper 

dans le navigateur.

Lorsqu'on relance le serveur avec la commande 

le serveur écoute maintenant sur le port 8080 et plus 8000, donc pour que la page s'affiche correctement il faudrait modifier l'URL dans le navigateur :

http://@IP:8080 . Si on garde le port 8000 la page ne s'affichera pas car aucun serveur n'écoute sur ce port. Le port 5432 est un port utilisé par PostgreSQL donc cela ne vas pas fonctionner car le port est déjà occupé.

Il est possible de lancer 2 serveurs Python sur deux ports différents car chaque serveur fonctionnera indépendamment. Son intérêt est de pouvoir héberger plusieurs sites en même temps sur la même machine.

Avec cette commande `curl -v http://localhost:8000/test.css` tapé dans le navigateur nous observons ce résultat :

```
* Host localhost:8080 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
*   Trying [::1]:8080...
* Established connection to localhost (::1 port 8080) from ::1 port 55098
* using HTTP/1.x
> GET /test.css HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/8.16.0
> Accept: */*
>
* Request completely sent off
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: SimpleHTTP/0.6 Python/3.13.9
< Date: Mon, 19 Jan 2026 14:43:11 GMT
< Content-type: text/css
< Content-Length: 36
< Last-Modified: Mon, 19 Jan 2026 14:41:22 GMT
<
body{
background-color: #f3b8b8;
}* shutting down connection #0
```

Extension	Content-Type
.txt	Text/plain
.html	Text/html
.css	Text/css
.png	Image/png
.jpg	Image/jpeg
.gif	Image/gif
.js	Application/javascript

Le type MIME permet au navigateur de savoir quel type de fichier est reçu. Le serveur reconnaît automatiquement le type MIME en fonction de l'extension du fichier. Pour notre fichier css il est envoyé avec le type `/test.css` et ça permet au navigateur d'interpréter correctement le contenu.

La journalisation permet au serveur web de garder une trace des requêtes reçues, des réponses envoyées, de la date, de l'heure et parfois même de l'adresse IP du client. Cela peut être utile pour diagnostiquer des erreurs, pour surveiller l'activité des serveurs, repérer des tentatives. Dans les lignes affichées par les résultats du dessus on retrouve souvent l'adresse IP du client la date, l'heure, la méthode http utilisée, la ressource demandée, la version http et le code de la réponse. Les codes vus sont 200 qui est la ressource

demandée existe et a été envoyée correctement ; le 404 qui indique que la ressource demandée n'existe pas et le 304 indique que le fichier n'a pas changer donc le navigateur peut utiliser sa version en cache.

La journalisation d'un serveur web permet donc d'enregistrer toutes les requêtes reçues et les réponses envoyées. On y vois plusieurs informations utiles et cela permet de surveiller l'activité du serveur, de détecter les erreurs et d'aider au diagnostic en cas de problème.