Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Hes·so

Téléinformatique – Ch. 13

HTTP

Vincent Magnin
vincent.magnin@hefr.ch

Objectifs

- Connaître la structure et la fonction d'un URL.
- Connaître la structure des messages HTTP, les méthodes ainsi que les codes de retour principaux.
- Comprendre une mesure Wireshark d'un échange HTTP.
- Connaître les différences et les bénéfices de HTTP/2.

Le World Wide Web

Internet relie des milliers de réseaux entre eux, qui comportent notamment des serveurs web accessibles au public. Ces serveurs hébergent les millions de sites qui existent.

Le World Wide Web (www) est un système hypertexte public fonctionnant sur Internet qui permet de consulter, grâce à un navigateur (*browser*), des pages accessibles sur des sites web.

Un hypertexte est un document contenant des unités d'information liées entre elles par des hyperliens.

Le World Wide Web constitue l'une des nombreuses applications d'Internet.

URI et URL

Un URI (Uniform Resource Identifier) est une chaîne de caractères utilisée pour identifier de manière unique une ressource, qu'elle soit située sur Internet ou ailleurs.

Une URI peut prendre différentes formes, notamment des URLs, des URN (Uniform Resource Names) et des URIs génériques.

Un URL (Uniform Resource Locator) est donc un type d'URI particulier. Il permet d'identifier une ressource réseau selon la norme suivante :

<méthode accès>://<station>[:<port>]/<chemin>/[<ressource>]

Le numéro de port est nécessaire si la valeur par défaut de la méthode d'accès n'est pas utilisée. Par exemple, la méthode d'accès *http* utilise le port par défaut 80.

HTTP

HTTP (HyperText Transfer Protocol) est un protocole de transfert simple. Il utilise le modèle C/S et est porté par le port 80/TCP.

Comme observé au chapitre TCP/UDP :

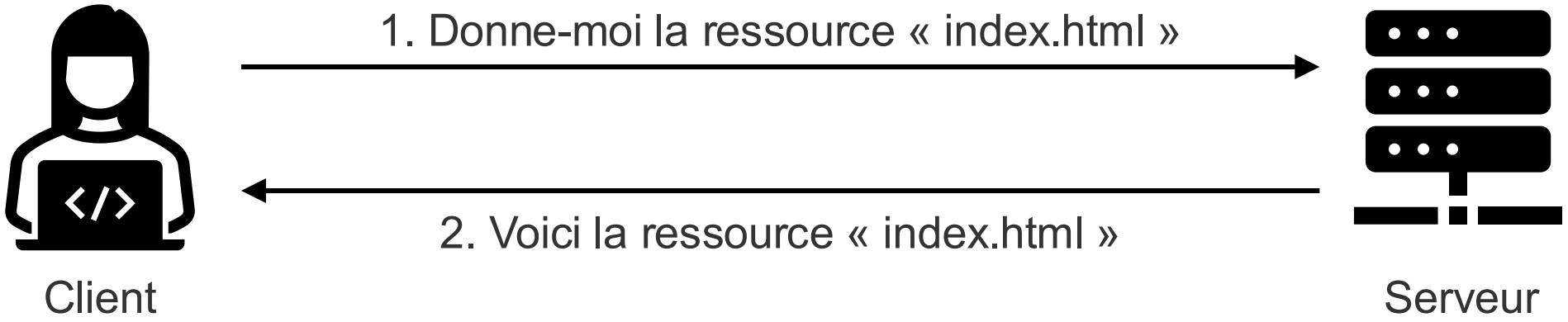
- Le port 80 est utilisé côté serveur.
- Le côté client utilise un port dynamique choisit par l'OS.

HTTP existe en plusieurs versions, et les plus utilisées aujourd'hui sont les versions 1.1 et 2.

Il existe également HTTPS, qui est la version sécurisée du protocole HTTP. Les données qui transitent par ce protocole sont chiffrées en utilisant le protocole TLS (Transport Layer Security). HTTPS utilise le port 443/TCP.

HTTP (2)

Comme le protocole HTTP suit le modèle C/S, les échanges sont relativement simples. Les messages HTTP sont envoyés sous format texte.



HTTP (3)

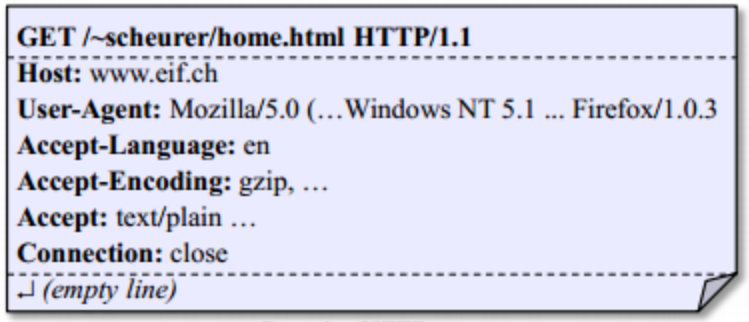
Le format des messages est très important dans le protocole HTTP.

Une requête doit avoir :

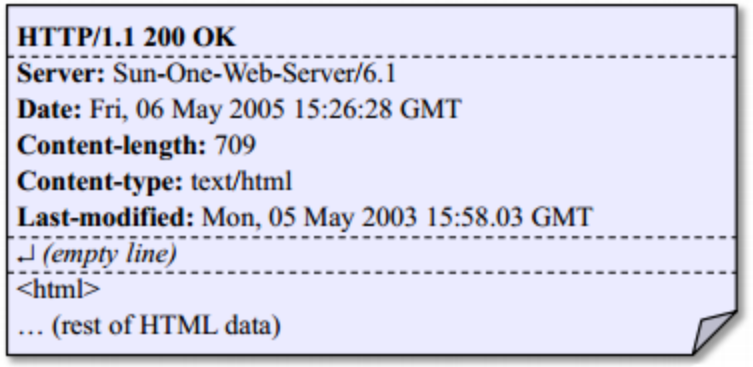
- Une *request* : type, URL, version HTTP.
- Un *header* : informations sur le navigateur, langue, formats de réponses acceptés, codage...
- Une *empty line* : séparation du header avec le body.
- Le *body* : généralement vide.

Une réponse doit avoir :

- Un *status* : version HTTP et code de retour de la réponse.
- Un *header* : inforations sur le serveur web, type de codage...
- Une *empty line* : séparation du header avec le body.
- Le *body* : la ressource qui a été demandée par le client (page HTML, image...).



Requête HTTP



Réponse HTTP

HTTP (4)

Time	Source	Destination	Protocol	Info
1.094163	160.98.101.195	208.97.189.107	HTTP	GET / HTTP/1.1
1.105397	208.97.189.107	160.98.101.195	HTTP	HTTP/1.1 200 OK (text/html)

```
⊟ Hypertext Transfer Protocol
  ⊞ GET / HTTP/1.1\r\n
    Host: www.perdu.com\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.8) Gecko/20100722 Firefox/3.6.8\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 115\r\n
    Connection: keep-alive\r\n
    \r\n
```

```
⊟ Hypertext Transfer Protocol
  ⊞ HTTP/1.1 200 OK\r\n
    Date: Wed, 25 Aug 2010 01:40:21 GMT\r\n
    Server: Apache\r\n
    Last-Modified: Tue, 02 Mar 2010 18:52:21 GMT\r\n
    ETag: "4bee144-cc-dd98a340"\r\n
    Accept-Ranges: bytes\r\n
    Vary: Accept-Encoding\r\n
    Content-Type: text/html\r\n
  ⊞ Content-Length: 163\r\n
    Connection: Keep-Alive\r\n
    Content-Encoding: gzip\r\n
    Age: 28367\r\n
    \r\n
    Content-encoded entity body (gzip): 163 bytes -> 204 bytes
⊞ Line-based text data: text/html
```

HTTP (5)

Il existe différents types de requêtes HTTP, qui sont utilisées dans différentes situations.

La méthode GET :

- Demande une ressource.
- Tous les paramètres sont visibles dans l'URL.

La méthode POST :

- Envoi de données vers une ressource.
- Tous les paramètres sont dans le message HTTP.

La méthode HEAD :

- Demande seulement l'entête HTTP de la ressource demandée.

Il existe de nombreuses autres méthodes, comme PUT, OPTIONS, DELETE, TRACE...

HTTP (6)

Les codes de retour HTTP servent au serveur à indiquer au client comment s'est passée le traitement de la requête. Le fichier demandé a-t-il été trouvé ? Une erreur est-elle survenue ?

Code	Signification	Exemple
1xx	Information	Peu utilisé aujourd'hui
2xx	Succès	200 = Request succeeded
3xx	Redirection	301 = Resource permanently moved (see location header) 302 = Resource temporarily moved (see location header)
4xx	Erreur client	400 = Syntax error in request 401 = Protected resource (authentication needed) 403 = Access forbidden 404 = Ressource not found
5xx	Erreur serveur	500 = Internal server error 501 = Request method not implemented 503 = Server busy, try again later...

HTTP/2

HTTP/2 est une version évoluée de HTTP/1.1. Les sites web d'aujourd'hui augmentent en complexité et le protocole HTTP/1.1 a du mal à suivre les évolutions. Le protocole a été publié en 2015, selon la RFC 7540 (HTTP/1.1 date de 1999). Un bon site de test pour HTTP/2 est http://www.http2demo.io/.

Les différences avec HTTP/1.1 sont les suivantes :

- Format binaire (traitement des entêtes plus efficient).
- Multiplexage des requêtes / réponses (gagne du temps).
- Le mode Server Push.
- Compression des entêtes.
- Priorisation des flux.
- Etc.

HTTP/2 (2)


HTTP/1.1 Baseline
1
Open Connection
2
GET /index.html
3
RESPONSE
4
GET /styles.css
5
RESPONSE
6
GET /scripts.js
7
RESPONSE
8
Client Renders Page
Connection Closed
9
Time
HTTP/2 Multiplexing
1
Open Connection
2
GET /index.html
3
RESPONSE
4
GET /styles.css
GET /scripts.js
5
RESPONSE
RESPONSE
6
Client Renders Page
Connection Remains Open
7

Références

- Ancien cours « Téléinformatique » (G. Waeber, S. Paccard, Q. Vaucher, N. Wirth).
- Cours « Systèmes d'information » (R. Scheurer)
- Ancien cours « Téléinformatique » (M. Roch-Neirey).