Protocole HTTP

GUINKO Tonguim Ferdinand

5 février 2025

Sommaire

1. Introduction
2. Requête HTTP
3. Réponse HTTP
4. URL et URI
5. Proxy HTTP
6. Cookies

Pour aller plus loin

- http://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- http://fr.wikipedia.org/wiki/Suite_des_protocoles_Internet

1 Introduction
2 Requête HTTP
3 Réponse HTTP
4 URL et URI
5 Proxy HTTP
6 Cookies

Généralités

- HTTP : HyperText Transfer Protocol, protocole de communication client serveur ;
- Protocole le plus utilisé sur Internet depuis 1990 ;
- Le but du protocole HTTP est de permettre un transfert de fichiers (essentiellement au format HTML) localisés grâce à une chaîne de caractères appelée URL entre un navigateur (le client) et un serveur Web ;
- HTTP est employé pour chaque transaction, il est présent dans chaque requête concernant un document ou un graphique du WEB, à chaque fois que l'on clique sur un lien hypertexte, et pour chaque formulaire soumis.

Généralités (Suite ...)

HTTP Fonctionne comme une combinaison de FTP et de SMTP :

- FTP : parce qu'il transfert des fichiers et utilise le service TCP
- SMTP : parce que les données transférées entre le client et le serveur sont semblables aux messages SMTP :
  - Le format des message contrôlés par des entêtes MIME-LIKE ;
  - Les messages HTTP sont délivrés immédiatement.

Généralités (Suite ...)

- Versions :
  - 0.9 : était uniquement destinée à transférer des données sur Internet (en particulier des pages Web écrites en HTML ;
  - 1.0 : permet de transférer des messages avec des en-têtes décrivant le contenu du message en utilisant un codage de type MIME ;
  - 1.1 :
- HTTP est un protocole ASCII (American Standard Code for Information Interchange). Il se situe au niveau 7 (Application) du modèle OSI, au même titre que FTP (File Transfer Protocol) ;
- HTTPS (avec S pour secured, soit « sécurisé ») est la variante du HTTP sécurisée par l'usage des protocoles SSL ou TLS.

Principe de fonctionnement de HTTP

HTTP est un protocole transactionnel, simple, basé sur le principe de Requête/Réponse. Il est dit sans état :

1. Le client envoie une requête, en spécifiant une méthode, des entêtes et des données s'il y a lieu, afin de récupérer un document en format HTML ou autre ;
2. Le serveur répond, indépendamment des requêtes précédentes et sans conserver la moindre information pour les requêtes à venir (Les serveurs peuvent cependant enregistrer les entêtes des requêtes à des fins statistiques et de déboggage) :
   a) Il renvoie une ligne de statut, des entêtes que le navigateur sera en mesure de reconnaître, suivi du document HTML ou autre demandé ; ensuite, il libère la connexion ;
   b) Le navigateur interprète alors la ligne de statut, les entêtes et affiche le document en fonction de la ligne de statut et des entêtes que le serveur Web a retournés.

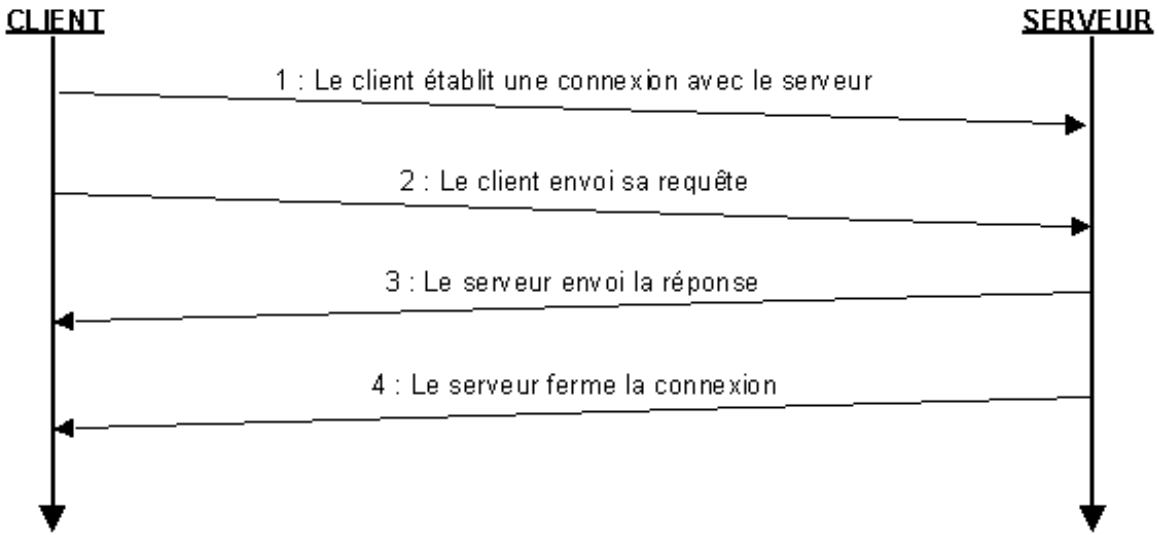
Principe de fonctionnement de HTTP (Suite ...)

Remarques :

1. HTTP n'est pas orienté session, il n'a aucun moyen d'identifier un utilisateur au cours d'une série de connexions à un site WEB ;
2. C'est le client qui ouvre la connexion, mais c'est le serveur qui la ferme. Une transaction HTTP correspond au transfert d'au plus une ressource. Ainsi, cela n'a pas de sens de dire que l'on est "connecté" à un site WEB. Les sites qui donnent à l'utilisateur l'illusion d'être connecté, utilisent entre autre, les cookies.

Principe de fonctionnement de HTTP (Suite ...)

Une transaction HTTP se décompose en quatre phases :



Principe de fonctionnement de HTTP (Suite ...)

Exemple pour la transaction :
http ://www.univ-ouaga.bf/Ibam/Dess2ITIC/protocoles/http/index.html

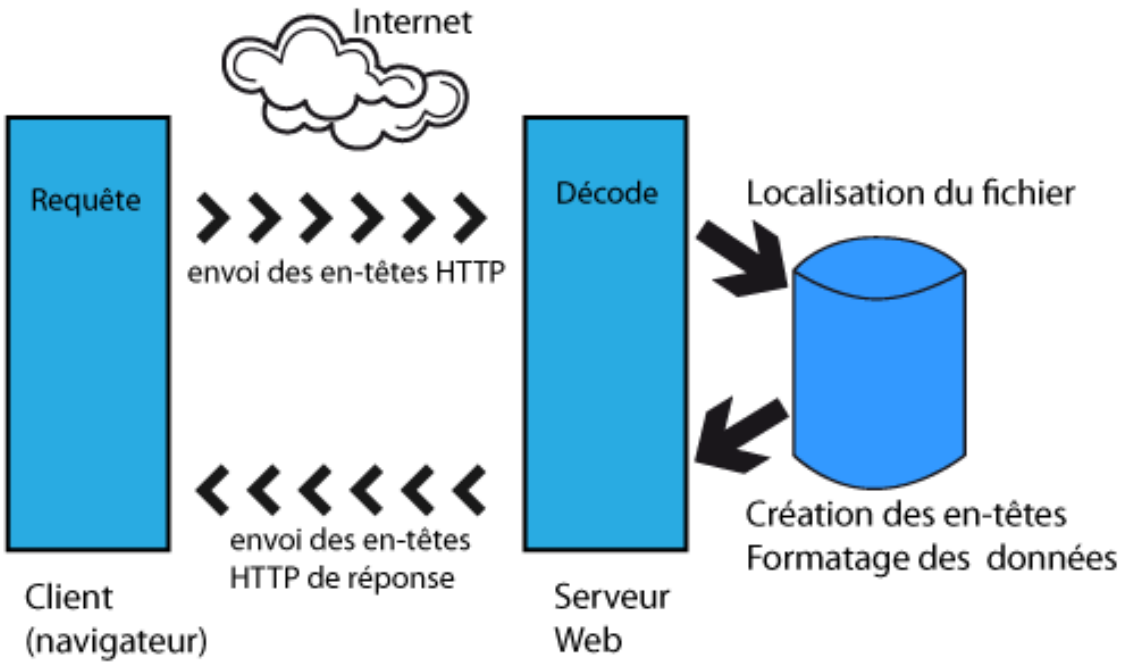
1. Le navigateur analyse l'URL et extrait le nom du serveur : www.univouaga.bf ;
2. Il demande au DNS l'adresse IP de la machine www.univ-ouaga.bf ;
3. Le DNS répond 212.52.131.9 ;
4. Le navigateur établit une connexion TCP sur le port 80 à l'adresse 212.52.131.9, puisqu'un port différent n'a pas été précisé dans l'URL ;

Principe de fonctionnement de HTTP (Suite ...)

Exemple pour la transaction :
http ://www.univ-ouaga.bf/Ibam/Dess2ITIC/protocoles/http/index.html (... suite)

1. Il envoie alors la commande GET /Ibam/Dess2ITIC_20122013/protocoles/http/index.html HTTP/1.1 ;
2. Le serveur www.univ-ouaga.bf envoie le fichier index.html ;
3. La connexion TCP est libérée par le serveur. La transaction est terminée ;
4. Le navigateur interprète et affiche la page correspondante à index.html ;
5. S' il y trouve des références à des images, sons, etc ..., il va faire une nouvelle transaction pour chacune d'entre elles et réactualisera son affichage au fur et à mesure.

Principe de fonctionnement de HTTP (Suite ...)



1 Introduction

2 Requête HTTP

3 Réponse HTTP

4 URL et URI

5 Proxy HTTP

6 Cookies

Syntaxe d'une requête

Requête HTTP : permet au navigateur de dialoguer avec le serveur Web.

Syntaxe :
Méthode URL Version_protocole
Entête : valeur
...
Entête : valeur
Données

Définition : méthode

Méthode :
Méthode est une commande spécifiant un type de requête, c'est-à-dire qu'elle demande au serveur d'effectuer une action. En général l'action concerne une ressource identifiée par l'URL qui suit le nom de la méthode.

Quelques méthodes

Table – Méthodes supportées par HTTP 1.1 :

Méthode	Description
GET	Méthode la plus courante pour demander une ressource. Une requête GET est sans effet sur la ressource, il doit être possible de répéter la requête sans effet.
HEAD	Ne demande que des informations sur la ressource, sans demander la ressource elle même.
POST	Utilisée pour ajouter une nouvelle ressource (un message sur un forum ou un article dans un site). L'URI fournie est l'URI d'une ressource liée à la nouvelle ressource (comme l'URI du forum ou site) et non l'URI de la ressource nouvellement crée.
OPTIONS	Permet d'obtenir les options de communication d'une ressource ou du serveur en général.

Quelques méthodes (... suite)

Table – Méthodes supportées par HTTP 1.1 :

Méthode	**Description**
CONNECT	Permet d'utiliser un proxy comme un tunnel de communication.
TRACE	Demande au serveur de retourner ce qu'il a reçu, dans le but de tester et effectuer un diagnostic sur la connexion.
PUT	Permet de remplacer ou d'ajouter une ressource sur le serveur. L'URI fourni est celui de la ressource en question.
DELETE	Permet de supprimer une ressource du serveur.

Remarque : Les méthodes les plus connues sont GET et POST.

Méthode GET

- Implications de l'envoi de champs par méthode GET :
  - Les clés/valeurs sont intégrées dans l'URL de la ressource demandée GET ressource HTTP/ver ; le protocole HTTP n'impose pas de limite pour la longueur des URLs. Cependant certains clients et serveurs HTTP limitent la taille des URLs.
  - Il est conseillé de ne pas dépasser 2048 caractères.
  - Conservation d'historiques d'URLs consultées par les clients et serveurs : implication sur la confidentialité des données
  - Rejeu possible de requêtes par navigation dans l'historique du client.

Méthode GET

- Dans quel cas utiliser la méthode GET pour l'envoi de champs ?
  - Pour des opérations sans effet de bord sur le serveur (nilpotence)
  - Pour communiquer des données de petite taille
  - Pour des informations peu confidentielles.

Méthode POST

Entêtes de requête

L'en-tête est composée de 3 parties :

1. En-tête générique qui concerne l'échange, la requête ou la réponse ;
2. En-tête de la requête qui concerne la requête elle-même ;
3. En-tête de l'entité qui concerne les données (les métas informations).

L'en-tête générique se compose principalement de la date et l'heure à laquelle on fait la requête (directive Date :).

Entêtes de requête (... suite)

Dans l'*en-tête de la requête*, on peut spécifier 5 choses :

Table – Méthodes supportées par HTTP 1.1 :

Entête	**Description**
From	Donne l'e-mail de la personne contrôlant le navigateur (cela peut poser des problèmes de respect de la vie privée).
Referer	URL de l'objet qui amène la requête (URL de la page où se trouve le lien).
User-Agent	L'identifiant du navigateur. Sert pour adapter la réponse au navigateur.
Authorization	Permet à un client de s'authentifier auprès du serveur.
IfModified-Since	Permet de spécifier que le document doit être envoyé s'il a été modifié depuis une certaine date. Permet de faire des GET conditionnels.

Entêtes de requête (... suite)

Table – Entêtes de requête

Entête	**Description**
Accept	Type de contenu accepté par le browser (par exemple text/html). Voir types MIME .
AcceptCharset	Jeu de caractères attendu par le browser
AcceptEncoding	Codage de données accepté par le browser.
AcceptLanguage	Langage attendu par le browser (anglais par défaut).
Content-Encoding	Type de codage du corps de la requête.

Entêtes de requête (... suite)

Table – Entêtes de requête

Entête	**Description**
Content-Language	Type de langage du corps de la requête.
ContentLength	Longueur du corps de la requête.
ContentType	Type de contenu du corps de la requête (par ContentType exemple text/html). Voir types MIME.
Date	Date de début de transfert des données.
Forwarded	Utilisé par les machines intermédiaires entre le browser et le serveur.
From	Permet de spécifier l'adresse email du client.

Entêtes de requête (... suite)

Table – Entêtes de requête

Entête	**Description**
IfUnmodified-Since	Permet de spécifier au serveur Web que l'on veut recevoir le document demandé seulement s'il n'a pas été modifié depuis une date précise.
Link	Relation entre deux URL.
OrigURL	URL d'origine de la requête.
UserAgent	Chaîne donnant des informations sur le client, comme le nom et la version du navigateur, du système d'exploitation.

1 Introduction

2 Requête HTTP

3 Réponse HTTP

4 URL et URI

5 Proxy HTTP

6 Cookies

Réponse du serveur : syntaxe

La réponse consiste en un ensemble de lignes envoyées au navigateur par le serveur. Parmi cet ensemble de lignes figure une **ligne de statut** sous forme de texte.

ligne de statut : ligne précisant la version du protocole utilisé et l'état du traitement de la requête à l'aide **d'un code** et d'un **texte explicatif**. Les codes sont ceux que l'on voit lorsque le navigateur n'arrive pas à nous fournir la page demandée. Le code de réponse est constitué de trois chiffres : le premier indique la classe de statut et les suivants la nature exacte de l'erreur. La ligne comprend trois éléments devant être séparés par un espace :

Réponse du serveur : syntaxe (... suite)

1. La version du protocole utilisée par le serveur ;
2. Le code de statut de la réponse sous forme numérique : ce sont les codes qu'affiche le navigateur lorsqu'il n'arrive pas à fournir la page demandée suite à une requête. Le code de réponse est constitué de trois chiffres : le premier indique la classe de statut et les suivants la nature exacte de l'erreur ;
3. La signification du code ; le code est constitué de trois chiffres : le premier indique la classe de statut et les suivants la nature exacte de l'erreur.

Code de statut

Table – Code de statut (... suite)

Code	Message	Description
10x	Message d'information	Ces codes ne sont pas utilisés dans la version 1.0 du protocole
20x	Réussite	Ces codes indiquent le bon déroulement de la transaction
200	Ok	La requête a été accomplie correctement
201	Created	Elle suit une commande POST, elle indique la réussite, le corps du reste du document est sensé indiquer l'URL à laquelle le document nouvellement créé devrait se trouver.
202	Accepted	La requête a été acceptée, mais la procédure qui suit n'a pas été accomplie
203	Partial information	Lorsque ce code est reçu en réponse à une commande GET, cela indique que la réponse n'est pas complète

Code de statut (... suite)

Table – Code de statut (... suite)

Code	Message	Description
204	No response	Le serveur a reçu la requête mais il n'y a pas d'information à renvoyer
205	Reset content	Le serveur indique au navigateur de supprimer le contenu des champs d'un formulaire
206	Partial content	Il s'agit d'une réponse à une requête comportant l'entête range. Le serveur doit indiquer l'entête *contentRange*

(Code de statut (... suite)

Table – Code de statut (... suite)

Code	Message	Description
30x	Redirection	Ces codes indiquent que la ressource n'est plus à l'emplacement indiqué
301	Moved	Les données demandées ont été transférées à une nouvelle adresse
302	Found	Les données demandées sont à une nouvelle URL, mais ont cependant peutêtre été déplacées depuis...
303	Method	Cela implique que le client doit essayer une nouvelle adresse, en essayant de préférence une autre méthode que GET
304	Not modified	Si le client a effectué une commande GET conditionnelle (en demandant si le document a été modifié depuis la dernière fois) et que le document n'a pas été modifié il renvoie ce code.

Code de statut (... suite)

Table – Code de statut (... suite)

Code	Message	Description
40x	Erreur due au client	Ces codes indiquent que la requête est incorrecte
400	Bad request	La syntaxe de la requête est mal formulée ou est impossible à satisfaire
401	Unauthorized	Le paramètre du message donne les spécifications des formes d'autorisation acceptables. Le client doit reformuler sa requête avec les bonnes données d'autorisation
402	Payment required	Le client doit reformuler sa demande avec les bonnes données de paiement
403	Forbidden	L'accès à la ressource est tout simplement interdit
404	Not found	Classique ! Le serveur n'a rien trouvé à l'adresse spécifiée. Parti sans laisser d'adresse... :)

Code de statut (... suite)

Table – Code de statut (... suite)

Code	Message	Description
50x	Erreur due au serveur	Ces codes indiquent qu'il y a eu une erreur interne du serveur
500	Internal error	Le serveur a rencontré une condition inattendue qui l'a empêché de donner suite à la demande (comme quoi il leur en arrive des trucs aux serveurs...)
501	Not implemented	Le serveur ne supporte pas le service demandé (on ne peut pas tout savoir faire...)
502	Bad gateway	Le serveur a reçu une réponse invalide de la part du serveur auquel il essayait d'accéder en agissant comme une passerelle ou un proxy

Code de statut (... suite)

Table – Code de statut (... suite)

Code	Message	Description
503	Service unavailable	Le serveur ne peut pas vous répondre à l'instant présent, car le trafic est trop dense (toutes les lignes de votre correspondant sont occupées veuillez rappeler ultérieurement)
504	Gateway timeout	La réponse du serveur a été trop longue visàvis du temps pendant lequel la passerelle était préparée à l'attendre (le temps qui vous était imparti est maintenant écoulé...)

1 Introduction

2 Requête HTTP

3 Réponse HTTP

4 URL et URI

5 Proxy HTTP

6 Cookies

URL

- URL : Uniform Resource Locator, littéralement "localisateur uniforme de ressource" ;
- Chaîne de caractères codé en ASCII utilisée pour adresser les ressources du World Wide Web (ressource physique ou abstraite) : document HTML, image, son, forum Usenet, boîte aux lettres électronique, etc.

Syntaxe :

protocole ://serveur[:port]/[chemin_acces]/fichier#position

URL (Suite ...)

Table – Syntaxe d'une URL

Protocole	Nom du protocole : le plus souvent HTTP
serveur	Nom d'une machine reliée à Internet (www.univ-ouaga.bf) ou son adresse (212.52.131.9)
[port]	Numéro de port sur lequel le serveur est en attente. Suivant le protocole utilisé, il existe toujours une valeur par défaut (80 pour HTTP)
[chemin]	Chemin (suite de répertoires séparés par des /) vers le document recherché
fichier	Nom du document recherché
[#position]	Nom désignant une position à l'intérieur du document

URL (Suite ...)

Table – Valeurs prises par **protocole**

Protocole	**Description**
file	Permet d'accéder aux fichiers locaux du poste de travail de l'usager
ftp	Permet d'accéder à un document sur un serveur FTP
gopher	Permet d'accéder à un document sur un serveur Gopher
http	Permet d'accéder à un document sur un serveur Web
mailto	Permet d'envoyer un courrier électronique
news	Permet d'accéder à un serveur Usenet
telnet	Permet d'accéder à un ordinateur via telnet

URL (Suite ...)

Exemples :

- http ://www.uqar.ca :8888
- file :///c :/temp/fichier_windows.html
- mailto :tonguim@uqar.ca

1 Introduction

2 Requête HTTP

3 Réponse HTTP

4 URL et URI

5 Proxy HTTP

6 Cookies

Proxy HTTP : définition

- Programme qui agit en tant qu'intermédiaire entre un client et un serveur ;
- Reçoit les requêtes des clients et les transmets aux serveurs concernés, et inversement ;
- Lorsqu'un client utilise un proxy, toutes ses requêtes sont transmises à ce proxy, plutôt qu'au serveur indiqués dans l'URL.

Proxy HTTP (Suite ...)

Exemple :

- supposons qu'une entreprise dispose d'un site web très achalandé et qu'elle souhaite optimiser l'utilisation de la bande passante dont elle dispose. Si certaines pages sont souvent accédées elles peuvent être "cachées". Préservation de la bande passante.
- certaines entreprises sont si larges qu'elles ne disposent pas de suffisamment d'adresses publiques pour chacun de leurs ordinateurs. S'il n'est pas possible d'attribuer une adresse publique à un client, ce dernier peut envoyer sa requête au serveur ; mais comment le serveur lui répondra t'il ? Solution : mettre en place un serveur de translation d'adresses, par exemple, à l'aide d'un proxy.

1 Introduction

2 Requête HTTP

3 Réponse HTTP

4 URL et URI

5 Proxy HTTP

6 Cookies

Généralités

Un cookie est une paire clé/valeur stocké par le client et associé à un domaine de validité.

- Le serveur demande l'installation d'un cookie par l'en-tête Set-cookie d'une réponse.
- Le client stocke le cookie et l'associe au domaine.
- Lorsque le client réémet une requête vers un domaine, il réenvoit l'ensemble des cookies stockés pour ce domaine par l'en-tête Cookie dans la requête.

Usage des cookies

- Utilité :
  - Conserver un identifiant de session :
    - Pour stocker un profil côté serveur
    - Pour servir d'authentifiant temporaire
  - Conserver des préférences de l'utilisateur côté client (fuseau horaire, devise préférée...)
- Limites des cookies :
  - Peu adapté pour conserver des données volumineuses
  - Validité globale pour un client : difficilement restreignable à une instance de navigation
    - Comment maintenir plusieurs paniers d'achat sur un même site dans deux fenêtres d'un même navigateur ?

Sécurité

- Risque de capture d'un identifiant de session contenu dans un cookie :
  -
  - Par espionnage d'une connexion non chiffrée (point d'accès WiFi public)
  - Par injection de code dans une page HTML (Cross Site Scripting XSS)
- Précautions :
  - Il faut toujours déspécialiser le contenu d'un formulaire renvoyé.
    - Soumission furtive d'un formulaire par un site tiers (Cross Site Request Forgery CSRF) :
  - Il faut ajouter à tout formulaire HTML un champ caché avec un jeton vérifié par le serveur
  - Utilisation de l'API de stockage local introduite par HTML5

Vie privée

- Super-cookies sur des domaines trop larges (normalement refusés par le client sur domaines avec "1 point" tels que .com, .fr... par contre .gouv.fr pourrait être accepté)
- Cookie de durée de vie trop importante
- Cookie tiers : intégration d'éléments de sites tiers sur une page avec envoi de cookie (image invisible...)
- Cookie zombie utilisant de façon détournée le cache ou l'historique du navigateur pour survivre
- Exploitation de la signature comportementale du navigateur (en-têtes HTTP, comportement canvas HTML5...)
- Cookie utilisant des greffons du navigateur (cookie Flash...)