

Fondamentaux

HTTP

## Table des matières

1.1 - Présentation.....	<a href="#">3</a>
1.1.1 - Définition.....	<a href="#">3</a>
1.1.2 - Syntaxes – en fonction des langages – pour requêter vers un serveur.	<a href="#">4</a>
1.1.3 - Schéma.....	<a href="#">5</a>
1.1.4 - Exemple.....	<a href="#">6</a>
1.2 - La structure d'une requête HTTP.....	<a href="#">7</a>
1.3 - La structure d'une réponse HTTP.....	<a href="#">8</a>
1.4 - Les entêtes HTTP.....	<a href="#">9</a>
1.5 - Comparaisons GET et POST.....	<a href="#">10</a>
1.6 - Cookies et variables de session.....	<a href="#">11</a>
1.6.1 - Cookie.....	<a href="#">11</a>
1.6.2 - Variable de session.....	<a href="#">11</a>
1.6.3 - Comparatif Cookie-Variable de Session.....	<a href="#">12</a>
1.7 - URI, URN, URL.....	<a href="#">13</a>

## 1.1 - PRÉSENTATION

### 1.1.1 - Définition

D'abord au lieu de parler de protocole HTTP il faudrait parler **de protocole de communication entre 2 logiciels**, ce serait plus clair pour des débutants en la matière.

Protocole est synonyme de règles. On se parle selon certaines règles !

Donc nous avons 2 logiciels : un logiciel « client » et un logiciel « serveur ».

Le **logiciel client** est le plus souvent un navigateur (Chrome, Firefox, IE, Opera, Safari, QupZilla, Midori, Epiphany, Lynx, Links, ...) installé sur une machine physique.

Le **logiciel serveur** est un serveur HTTPd (d pour daemon) – dénommé aussi serveur Web - installé sur une autre machine physique. Les principaux serveurs HTTP sont Apache, IIS, Glassfish, NGINX, ...

Les **principales méthodes** – de requêtes – sont GET, HEAD, POST, PUT et DELETE.

Le protocole HTTP est un protocole sans état (**StateLess**).

Ce qui signifie que c'est un protocole qui traite chaque demande comme une transaction indépendante étrangère à toute demande précédente. La communication se compose de paires indépendantes de demande et de réponse. Le serveur n'a pas à conserver des informations de session ou l'état de chaque partenaire de communication concernant les demandes multiples.

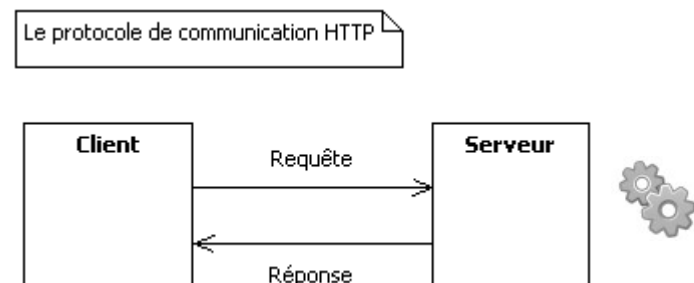
Pour pallier ce manque de gestion d'état on utilise les cookies et/ou les variables de session (cf plus loin).

**1.1.2 - Syntaxes – en fonction des langages – pour requêter vers un serveur**

Langage	Mode
HTML	<form action="url" ...> <a href="url" ... /> <meta http-equiv="refresh" content="0; URL=url">
JavaScript	document.location.ref = "url"; Requête AJAX
PHP	header("location: url");
JavaEE	response.sendRedirect("url"); response.setHeader("refresh", "url"); getServletContext().getRequestDispatcher("url").forward(request, response);  <jsp:forward page="url" />

### 1.1.3 - Schéma

Le client envoie une requête,  
le serveur traite la requête,  
le serveur envoie une réponse.



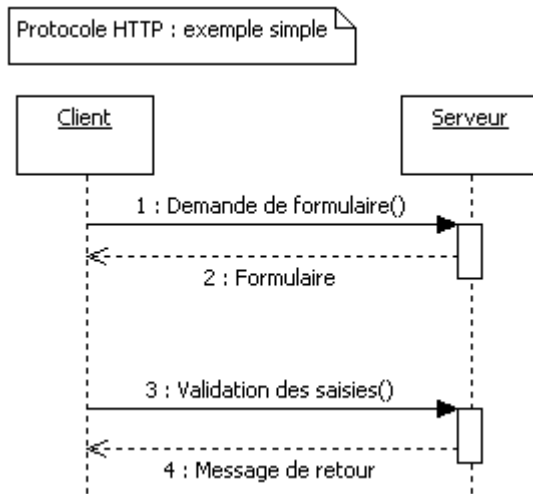
Note :

la plupart du temps la requête est une demande de page et la réponse est une page web,  
mais la réponse peut être une 404 (ressource introuvable),  
ou un 505 (erreur interne du serveur),  
etc.

La requête peut-être une requête AJAX et dans ce cas on ne requête pas vers une page mais vers une servlet ou un code PHP, ..., qui renvoie du texte ou du XML ou du JSON ou du JSONP ou du JavaScript.

### 1.1.4 - Exemple

Un formulaire de saisies ...



## **1.2 - LA STRUCTURE D'UNE REQUÊTE HTTP**

Ligne de commande (Méthode, URL, Version de protocole)

En-tête de requête

<nouvelle ligne>

Corps de requête

Notes :

lors d'une requête GET les données sont dans la ligne de commande et apparaîtront dans la barre d'URL sont la forme ?param1=valeur1&param2=valeur2 ...

Lors d'une requête POST les données sont dans le corps de la requête. Elles n'apparaîtront pas dans la barre d'URL.

GET http://www.commentcamarche.net HTTP/1.0

Accept : text/html

If-Modified-Since : Saturday, 15-January-2000 14:37:11 GMT

User-Agent : Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)

cf plus loin les détails sur les en-têtes HTTP.

## 1.3 - LA STRUCTURE D'UNE RÉPONSE HTTP

Ligne de statut (Version, Code-réponse, Texte-réponse)  
En-tête de réponse  
<nouvelle ligne>  
Corps de réponse

HTTP/1.0 200 OK  
Date : Sat, 15 Jan 2000 14:37:12 GMT Server : Microsoft-IIS/2.0  
Content-Type : text/HTML  
Content-Length : 1245  
Last-Modified : Fri, 14 Jan 2000 08:25:13 GMT

cf plus loin les détails sur les en-têtes HTTP.



## 1.4 - LES ENTÊTES HTTP

Nous venons de voir que les requêtes et les réponses contiennent des en-têtes. Voici une liste non exhaustive d'en-têtes HTTP.

En-tête	Description	Requête	Réponse
Accept	Content-Types that are acceptable for the response		
Accept-Charset	Character sets that are acceptable (UTF-8, ...)		
Accept-Encoding	List of acceptable encodings (gzip, ...)		
Accept-Language	List of acceptable human languages for response (en-US, ...)		
Cookie	An HTTP cookie previously sent by the server		
Host	The DOMAIN NAME of the server and the TCP port number		
User-Agent	The user agent string of the user agent		
Etc			

## 1.5 - COMPARAISONS GET ET POST

	GET	POST
Sémantique	Obtenir	Envoyer
Effet sur les ressources serveur	Aucune	Modification
URL	Données présentes	Données absentes
Corps de la requête	Données absentes	Données présentes
Sécurité	Peu sûr	Plus sûr
Quantité de données	Limitée (jusqu'à 8000 octets ça passe avec Chrome mais la norme est 1024 octets)	Illimitée
Type de données	Texte	Texte et binaire

Qui peut le plus, peut le moins !

## **1.6 - COOKIES ET VARIABLES DE SESSION**

### **1.6.1 - Cookie**

Un *cookie* est une petite quantité d'informations – 4 ko sous forme d'une chaîne de caractères - envoyée par le serveur au navigateur et stockée sur le poste du client. Cf plus de détails dans le tableau comparatif.

### **1.6.2 - Variable de session**

Les variables de session permettent de stocker des informations concernant l'utilisateur côté serveur et sont disponibles dans toutes les pages du site pendant la durée de la session (30 minutes par défaut). Les variables de session sont stockées sur le disque dur du serveur.

La session persiste pour une période de temps spécifiée. Elle est aussi détruite lorsque l'utilisateur ferme le navigateur.

Le serveur peut maintenir une session de plusieurs façons ; soit au moyen d'un cookie soit par réécriture d'URL.

### 1.6.3 - Comparatif Cookie-Variable de Session

	Cookie	Session
Type	Chaîne de caractères.	Tout type d'objet (cela peut dépendre d'un langage serveur à un autre).
Taille	4 ko par cookie. 50 cookies par domaine.	Pas de limite.
Quantité	Limitée.	Illimité ou presque.
Stockage	Client (sur le DD) par le navigateur. Les cookies transitent dans les en-têtes de requête et de réponse.	Serveur (sur le DD).
Durée de vie	Jusqu'à la fermeture du navigateur ou jusqu'à une date précisée par le code.	La durée de la session (en général limitée à 30 minutes).

Note : avec le Storage d'HTML 5 les limites des cookies et certains problèmes de sécurité sont résolus et révolus.

## 1.7 - URI, URN, URL



### URI :

un URI, Uniform Resource Identifier (identifiant uniforme de ressource) est une chaîne de caractères identifiant une ressource sur un réseau (par exemple une ressource Web) physique ou abstraite, et dont la syntaxe respecte une norme d'Internet mise en place pour le World Wide Web (RFC 3986).

Un URI doit permettre d'identifier une ressource de manière permanente, même si la ressource est déplacée ou supprimée.

### URN :

un URN, Uniform Resource Name (nom uniforme de ressource), est le nom d'un standard informatique dans le domaine de l'Internet qui concerne principalement le World Wide Web.

Exemple : URN:ISBN:0-395-36341-1

est un numéro de l'International Standard Book Number (ISBN), qui fait référence à un livre, mais ne suggère ni où, ni comment en obtenir une copie réelle.

### URL :

un Uniform Resource Locator (localisation uniforme de ressource) est un URI qui, outre le fait qu'il identifie une ressource sur un réseau, fournit les moyens d'agir sur une ressource ou d'obtenir une représentation de la ressource en décrivant son mode d'accès primaire ou « emplacement » réseau.

Exemple : <http://localhost/index.html>

est un URL qui identifie une ressource (page d'accueil) et implique qu'une représentation de cette ressource (une page HTML en caractères encodés) peut être obtenue via le protocole HTTP.

Autres exemples :

<file:///opt/lampp/htdocs/algos/html/Accueil.html>

<ftp://p.b:mdp1234@ftpperso.free.fr:21>