

# Internet et le protocole HTTP

## Table des matières

|   |   |
|---|---|
| PRÉSENTATION.....   | 2 |
| OBJECTIF.....   | 2 |
| LE PROTOCOLE HTTP.....                                      | 2 |
| LA NOTION D'URL ET D'URI (Uniform Resource Identifier)..... | 3 |
| LES REQUÊTES HTTP.....                                      | 4 |
| LES VARIABLES DU PROTOCOLE HTTP.....                        | 5 |
| LES RÉPONSES DU SERVEUR OU CODE DE STATUS.....              | 6 |
| ÉTUDE D'UN EXEMPLE.....                                     | 7 |
| LES TYPES MIME.....   | 8 |
| EXEMPLE APPLICATIF.....                                     | 8 |

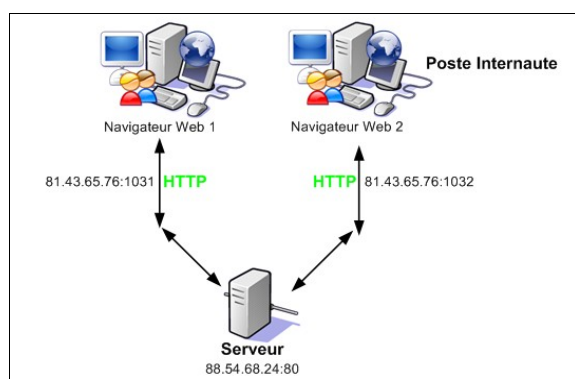
## PRÉSENTATION

### OBJECTIF

Lors de l'utilisation d'Internet, l'application cliente (le navigateur, browser ou butineur) située sur votre machine passe une requête à un serveur HTTP également appelé serveur Web.

Le navigateur envoie une demande à l'aide d'un couple **adresse IP:Port** sous la forme : 80.25.54.56:80.

Le numéro de port est tout aussi indispensable que l'adresse IP de la machine, il sert à aiguiller les informations échangées vers les bonnes applications.



Dans l'exemple ci-dessus, les numéros de ports 1031 et 1032 permettent d'aiguiller le retour des informations. Les informations sont alors affichées dans les applications concernées. Ces numéros de ports sont tirés au hasard et supérieurs à 1024. Les applications serveurs ont des numéros de port réservés. Lorsque nous consultons un serveur Web sans préciser de numéro de port, nous utilisons le port HTTP et le numéro 80.

### LE PROTOCOLE HTTP

L'ensemble du protocole HTTP 1.1 est décrit dans la RFC2616 (Request For Comment ou appel à commentaire) géré par l'organisme IETF (Internet Engineering Task Force). Les échanges entre les serveurs Web et les clients sont régis par le protocole HTTP. Ce protocole est indépendant du système d'exploitation utilisé (Windows, Linux, Mac, Android...). Le protocole HTTP se situe au niveau 7 (Application) du modèle OSI et lorsqu'un client envoie une requête vers le serveur, celui-ci répond de façon appropriée au client.

Le protocole HTTP (ensemble de règles et de conventions comme le protocole du langage français) (HyperText Transfert Protocol) régit les échanges entre les clients Web (Internet Explorer, GoogleChrome, FireFox, Safari, Opera...) et les serveurs Web (Apache, IIS, GlassFish, Tomcat...).

L'application cliente envoie des requêtes en mode texte au serveur. Les requêtes ont le format suivant :

**[type de requête] [page demandée et présente sur le serveur] [type de protocole] [options]**

Les navigateurs utilisent ce procédé lorsque nous saisissons une page dans l'URL d'un navigateur. Voici un exemple de requête HTTP:  
*GET /index.html HTTP/1.0*

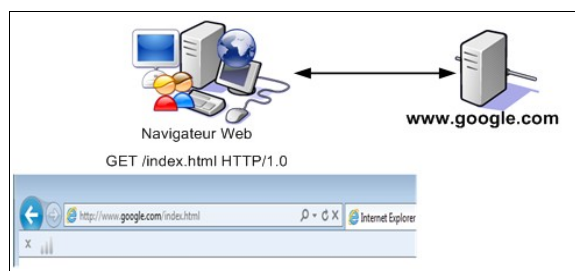
Le client souhaite récupérer (méthode GET) la page *index.html* du serveur et précise que la requête est de type HTTP version 1.0. En réponse, le serveur doit renvoyer au client le contenu de la page *index.html*.

### **LA NOTION D'URL ET D'URI (Uniform Resource Identifier)**

Sur Internet les informations sont des ressources désignées par identifiants appelés URI. Ces URI sont composées de deux sous-ensembles que sont les URL et URN. Le fait de cliquer sur une URI indique au navigateur que nous souhaitons utiliser une représentation de la ressource désignée. Le navigateur utilise le protocole HTTP et envoie une requête HTTP de type GET sur le port 80 du serveur désigné. En retour, le serveur lui renvoie la représentation actuelle de la ressource distante.

La définition d'URI est libre et nous pouvons les choisir à notre souhait. Cependant, il existe deux techniques de construction des URI :

- Par hiérarchie : la méthode par hiérarchie consiste à définir le chemin de l'URI à partir des différents niveaux de priorité de la ressource. Les éléments les plus prioritaires se trouvent à gauche de l'URI et les moins importants à droite. Ainsi, l'URI suivante permet de retourner les informations de l'aide d'identifiant 1 : <http://localhost:8080/webcole-war/rest/aide/1>.
- Par construction : la méthode par construction est très utilisée sur Internet et consiste à passer les paramètres dans l'URI à l'aide des délimiteurs ? et &. Ainsi, l'URI précédente pourrait avoir son équivalent suivant en mode par construction : <http://localhost:8080/webcole-war/rest?aide=1>. Cette représentation peut également utiliser le mode matrice : <http://localhost:8080/webcole-war/rest;aide=1>.



Pour bien comprendre le mécanisme, nous allons utiliser un client réseau permettant d'envoyer des commandes. La console Telnet permet d'envoyer des requêtes en mode texte sur un port désigné. Pour cela, nous utilisons les commandes suivantes sous Windows :

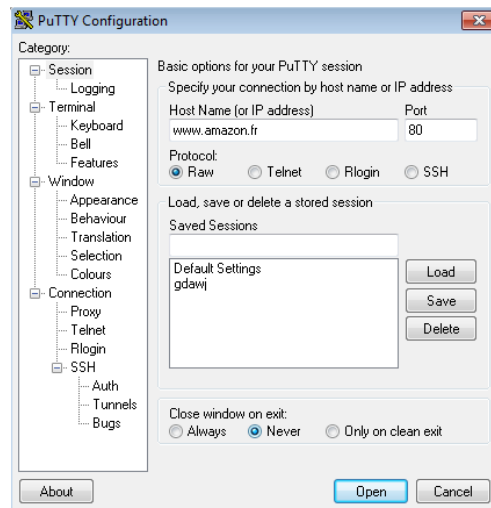
- > *Menu démarrer/exécuter et entrez* `pkgmgr /iu:TelnetClient`
- > *Une console graphique s'ouvre alors et nous permet de lancer des commandes textuelles.*

Le client réseau Putty permet également de réaliser des requêtes Telnet afin de vérifier les retours serveur sans interprétation par un browser.

Pour cela, il est nécessaire de lancer Putty et de préciser les informations suivantes :

- Host Name ou IP: le nom du serveur cible dans notre exemple [www.amazon.fr](http://www.amazon.fr).
- Port: le port d'écoute du serveur Web (généralement 80).

- Connection Type: Raw (afin de pouvoir passer en paramètre notre numéro de port 80)
- Cocher le bouton Close window on exit: Never (sinon, le retour serveur ne sera pas affiché dans la console).



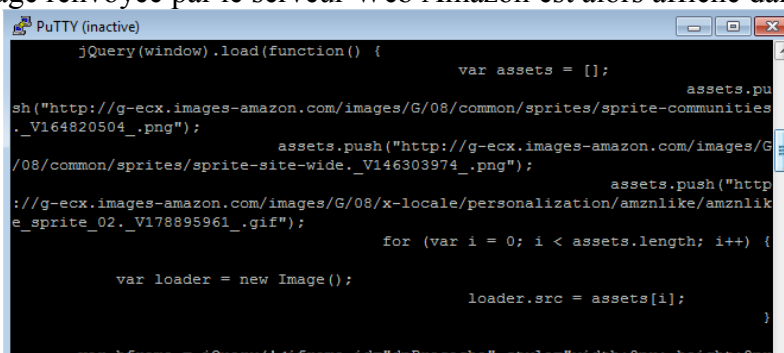
Lorsque la connexion est ouverte (bouton Open), une fenêtre de type console s'ouvre et nous pouvons alors saisir nos commandes HTTP.

Exemple pour la page d'accueil du site : GET / HTTP/1.0  
suivi de deux fois la touche Enter (CTRLF).



Nous allons pouvoir entrer à la main ce que le navigateur de votre choix passe au serveur lorsque nous entrons l'adresse <http://www.amazon.fr>. La réponse renvoyée est toujours au format SGML (XHTML, XML, HTML). Le rôle du navigateur est alors d'afficher correctement le code HTML et de réaliser une mise en page propre et élaborée (affichage des polices de caractères, des tableaux, des images...).

Le contenu de la page renvoyée par le serveur Web Amazon est alors affiché dans la console Putty.



## LES REQUÊTES HTTP

Les requêtes ou demandes des clients vers une adresse se font avec ce que l'on appelle des méthodes. Voici la liste des principales requêtes utilisées par le protocole HTTP :

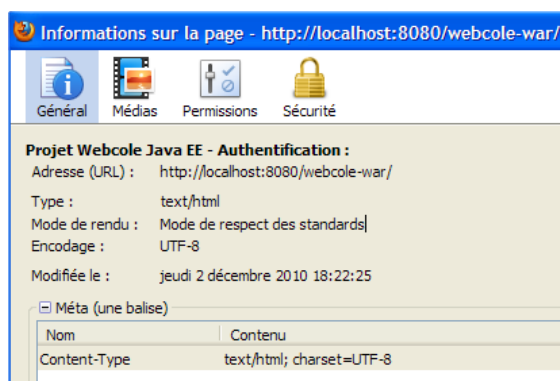
|     |   |
|-----|---|
| GET | Demande d'un document (le document est précisé en paramètre sinon index.html) |
|-----|---|

|         |  |
|---------|--|
| HEAD    | Demande l'entête d'une réponse HTTP (le début du document).  |
| POST    | Envoie un contenu au serveur (un formulaire, des données textuelles...). Un script serveur traite le contenu envoyé. |
| PUT     | Demande au serveur de modifier la ressource envoyée.   |
| DELETE  | Demande au serveur d'effacer une ressource.  |
| OPTIONS | Liste les options possible pour une ressource.   |

Lors de l'exécution d'une requête HTTP par un navigateur, la méthode HTTP apparaît forcément en premier. Après cette méthode, nous devons indiquer l'URI (Uniform Resource Identifier) concernée. L'URI peut être un chemin vers une page HTML ou une URL complète. Enfin, nous terminons cette ligne par la version du protocole utilisé en suivant la syntaxe : HTTP/1.X ou X représente la version du protocole utilisé.

*GET [www.amazon.fr/index.html](http://www.amazon.fr/index.html) HTTP/1.1*

➡ **Remarque :** Avec le navigateur Firefox et son plug-in WebDeveloper ou FireBug, il est possible d'afficher de façon claire les informations d'une réponse HTTP en utilisant le menu **Information – Informations sur la page**.



## LES VARIABLES DU PROTOCOLE HTTP

Lorsque le client navigateur envoie des requêtes à l'aide des méthodes HTTP (GET, POST...), celui-ci complète le contenu de la demande avec une liste de variables et de valeurs associées. Ces variables et leurs valeurs associées sont utilisés aussi bien par le client et le serveur lors des échanges. Ces variables permettent au client et au serveur de se comprendre.

Voici un tableau non exhaustif des variables couramment utilisées.

|                 |  |
|-----------------|--|
| Accept          | Spécifie les types de média acceptés pour la ressource demandée (text/html, image/jpeg...). Ces types sont appelés des types MIME. |
| Accept-Charset  | Spécifie l'encodage de la page réponse (UTF-8, français...).   |
| Accept-Language | Identique à Accept-Charset mais avec une limitation de l'encodage.   |

|               |  |
|---------------|--|
| Authorization | Utilisé par un client pour s'authentifier auprès du serveur.       |
| Connection    | Indique des paramètres particuliers pour la connexion.             |
| Cache-Control | Définit une politique de cache pour la ressource demandée.         |
| Age           | Utilisé pour la date du cache.                                     |
| Content-Type  | Précise le type de média du corps de la ressource (text/html).     |
| Date          | Date à laquelle le message est envoyé.                             |
| Host          | Précise le serveur virtuel sur lequel la requête est envoyée.      |
| Last-Modified | Renvoie la date de dernière modification de la ressource.          |
| Server        | Indique le serveur qui répond à la requête.                        |
| User-Agent    | Indique le type de client utilisé pour la requête (le navigateur). |

## LES RÉPONSES DU SERVEUR OU CODE DE STATUS

Le serveur renvoie la version du protocole utilisé suivie d'un code de retour qui correspond au status du serveur (à l'état du serveur).

Par exemple : HTTP/1.1 **200 OK**

Ces codes de status sont groupés en cinq catégories.

- Les codes de la catégorie 1XX servent à donner une réponse provisoire.
- Les codes de la catégorie 2XX sont utilisés lorsque la requête a été correctement reçue, comprise et traitée par le serveur.
- Les codes de la catégorie 3XX sont utilisés lorsqu'il faut procéder à une autre requête pour accéder à la ressource. A priori, le serveur sait que la ressource est accessible.
- Les codes de la catégorie 4XX sont utilisés lorsque une requête est syntaxiquement incorrecte ou incomprise par le serveur.
- Les codes de la catégorie 5XX sont utilisés pour préciser que la requête client est correcte mais que le serveur ne peut la satisfaire (il ne sait pas comment répondre).

Voici une liste non exhaustive des principaux codes en fonction de leur catégorie :

|     |                     |  |
|-----|---------------------|--|
| 100 | Continue            | Le client doit continuer à attendre la réponse à sa requête. Celle-ci est longue à exécuter.         |
| 101 | Switching Protocols | Le serveur va répondre en utilisant une autre version du protocole (ex : HTTP1.1 au lieu de HTTP1.0) |
| 200 | OK                  | La réponse est correctement renvoyée par le serveur. La requête est satisfaite.                      |
| 202 | Accepted            | Requête acceptée mais non traitée par le serveur.  |
| 204 | No Content          | La requête est satisfaite mais le serveur n'a rien de particulier à renvoyer.                        |
| 300 | Multiple Choices    | La ressource demandée existe sous plusieurs formats (ex : index.htm, index.html).                    |
| 301 | Moved Permanently   | La ressource demandée a changé d'emplacement. La directive   |

|     |                       |   |
|-----|-----------------------|---|
|     |                       | Location contient l'adresse de la nouvelle URI à contacter.   |
| 303 | See Other             | Proche du code de status 301 et demande une nouvelle requête GET pour obtenir la ressource.   |
| 304 | Not Modified          | Ce code est renvoyé lorsque le client effectue une requête GET conditionnelle et que le document demandé n'a pas été modifié depuis la date indiquée. |
| 400 | Bad Request           | La requête est syntaxiquement incorrecte (ex : pas de .fr, pas de .com).  |
| 401 | Unauthorized          | L'utilisateur doit s'authentifier pour accéder à la ressource.  |
| 403 | Forbidden             | Le serveur ne veut pas délivrer la ressource. Le client demandeur n'est pas authentifié.  |
| 404 | Not Found             | La ressource spécifiée est introuvable (erreur d'URL).  |
| 500 | Internal Server Error | Le serveur a eu un problème pour répondre à la demande. Souvent c'est une erreur du code exécuté sur le serveur.                                      |
| 501 | Not Implemented       | Le serveur ne peut pas exécuter la requête.   |
| 503 | Service Unavailable   | Le serveur ne peut satisfaire la requête mais il devrait pouvoir y répondre plus tard.  |

## ÉTUDE D'UN EXEMPLE

Le code présenté ci-après présente la requête du client suivie de la réponse du serveur au client.

### 1. requête du client (mode console, navigateur...)

```
GET /index.html HTTP/1.1
Accept : image/gif, image/jpeg, */*
Accept-language : fre
Accept-Encoding : gzip
User-Agent : Mozilla/4.0, Android
Host : ste-marie.fr
Connection : Keep-Alive
```

### 2. La réponse du serveur

```
HTTP/1.1 200 OK
Date : Mon, 04 Nov 2011 19:02:12 GMT
Server : Apache/2.0.1 (Debian/linux)
Last-Modified : Wed, 25 Sep 2011 11:23:34 GMT
Accept-Ranges : bytes
Content-Length : 7485
Connection : close
Content-type:text/html
<html>
<head>
<title>Le site Internet de l'école</title>
<body>
...
</body>
</html>
```

Quelle est la version du protocole utilisé ?

Que demande le client au serveur ?

Comment se nomme le serveur Web ?

A quelle date (et heure) a été renvoyée la réponse du serveur ?

Quelle est la date de dernière mise à jour du contenu ?

Quel est le type de média utilisé pour encoder la réponse du serveur ?

3. Le même client envoie peu après la requête suivante au serveur.

```
GET / HTTP/1.1
Accept:/*/*
Accept-language:fr
If-Modified-Since:Wed, 25 Sep 2011 11:23:34 GMT
User-Agent:Mozilla/4.0
Host:ste-marie.fr
Connection:Keep-Alive
```

4. La réponse envoyée par le serveur est la suivante.

```
HTTP/1.1 304 Not Modified
Date : Mon, 04 Nov 2011 19:05:01 GMT
Server : Apache/2.0.1 (Debian/linux)
Connection : close
```

Quelles sont les deux directives de conditions associées au GET émis par le client ?

A quoi correspond le code status renvoyé par le serveur ?

A quelle date et heure a été renvoyée la réponse du serveur ?

## ***LES TYPES MIME***

L'acronyme MIME signifie Multipurpose Internet Mail Extensions. Les types MIME ont été utilisés à l'origine pour joindre autre chose que du texte aux emails. Ces types servent à préciser avec quelle application doit être ouvert un contenu associé à un document et renvoyé par le serveur Web.

La déclaration d'un type MIME se décompose en deux parties :

**catégorie/type de données**

Exemple : image/jpeg (catégorie des images et contenu au format jpeg).

Le client (navigateur) peut annoncer avec la directive Accept quels sont les types MIME supportés (ou avec \*/\* qu'il les accepte tous). De son côté, le serveur envoie le type MIME du contenu qui va suivre (ex : Content-Type:text/html). Ainsi, le client pourra alors interpréter de façon correcte les données reçues sachant qu'un contenu correspond à une requête sur Internet.

Ces types MIME sont indispensables pour le bon fonctionnement des échanges HTTP. Ces types sont à l'origine de nombreux trous de sécurité sur les machines clientes.

Voici une liste non exhaustive des principaux types MIME utilisés sur Internet :

```
application/pdf pdf
application/powerpoint ppt
application/x-javascript js
application/mpeg mpga mp2 mp3 mp4
application/gif gif
application/jpeg jpeg jpg jpe
text/css css
text/html html htm
text/plain txt
text/xml xml dtd
```

## ***EXEMPLE APPLICATIF***


L'impression d'écran ci-dessous présente un site pirate très élaboré sur Internet. Le site était disponible à cette adresse : <http://www.lo-fstlne.com>. Le site réel est disponible à cette adresse :



<http://www.cargoline.de>.

L'extrait ci-dessous est tiré d'un bon de commande présenté sur le site dans l'interface de suivi.

Tracking results <http://www.lo-fastlane.com/track200891705210.htm>


Home | Contact

Flash Off

ACCUEIL

NOUVELLES

PARTENAIRES

PRODUITS

TRACK & TRACE

Track & Trace®

SafeTrade Service

### TRACKING INFORMATION

**Numéro de suivi 200891705210**

| Exportateur: |                   | Importateur: |                        |
|--------------|-------------------|--------------|------------------------|
| Nom:         | Rodney Russell    | Nom:         | Jerome Lafosse         |
| Adresse:     | 18 Rolis Crescent | Adresse:     | 220 Chemin de la Poste |
| Ville:       | Manchester        | Ville:       | Pannessières           |
| Code postal: | M15 5JY           | Code postal: | 39570                  |
| Pays:        | Royaume-Uni       | Pays:        | France                 |

#### L'INFORMATION DE PAQUET/RÉCIPIENT :

|  |   |
|--|---|
| Nombre de paquets:                         | *3  |
| Type De Paquet:                            | Double Safe Box   |
| Dimensions:                                | Standard  |
| Temps d'expédition estimé:                 | Entre 3- 4 Jours*   |
| Type de service:                           | *SafeTrade Service  |
| Instructions Spéciales pour la Livraison : | *Période de vérification de dix jours, à compter de la date d'arrivée |
| Numéro de Suivi:                           | 200891705210  |
| Description du produit:                    | 2012 Cannondale Super Six Evo Di2                                     |
| Valeur:                                    | Euro 2150   |
| Montant SafeTrade:                         | Euro 2150   |
| Assurance:                                 | *Oui  |
| Valeur à assurer:                          | Euro 2150   |

**Date d'expédition:** 30-08-2012

**Date estimée d'arrivée:** 3 Jours après la confirmation du paiement

**Statut:** SafeTrade Service - Etape 2

\*Attendons la confirmation de paiement

Étape 1 de la transaction SafeTrade a été achevée. Première étape comporte la réception et l'inspection de l'article. Nos techniciens ont inspecté l'article, vérifié ses documents et fait l'assurance pour le transit.

L'étape 2 de la transaction a été commencée. Afin d'achever l'étape 2, l'acheteur doit faire le transfert d'argent à l'agent de notre entreprise à Londres, RU.

**Montant total à transférer: € 2150**

Le paiement doit être adressé à notre chef de branche à Londres, RU. Vous trouverez en dessous de la page les détails de notre compte bancaire. Vous pouvez envoyer l'argent par virement bancaire.

S'il vous plaît, utilisez les coordonnées bancaires suivantes pour faire un virement bancaire en contactant votre banque ou par transfert de compte en ligne:

Détails de compte bancaire :

Nom de compte : - GURMAN TYMON

Nom de Banque - BARCLAYS BANK PLC

Account No. - 73053178

Sort Code: 205274

IBAN - GB65 BARC 2052 7473 0631 78