Université Nice SOPHIA ANTIPOLIS

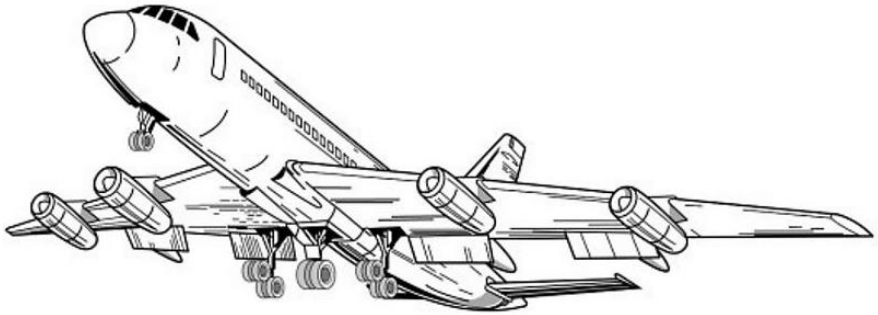
PLAN du Cours

- ✓ Du Web aux Web services
  - Le protocole commun HTTP
  - Les Web Services REST
- ✓ Les Web services
  - WS REST et SOAP
  - Format de représentation de données (SOAP, JSON, XML, ...)
  - Protocoles d'échanges : Requête-Réponse, 1 requête – N Réponse, One way message, ...
  - Contrats, Langages de description et Web Service (WSDL, WALD, ...)
- ✓ Des Web Services aux Services
  - Modèle récapitulatifs des WS
  - Changements de Binding
- ✓ Intro à Windows Communication Fundation (WCF)

Dans tous les cas : le Protocole HTTP

HyperText Transfert Protocol

HTTP 1.0 : RFC 1945
HTTP 1.1 : RFC 2616



Introduction

Les Principes et Eléments de base du Protocole

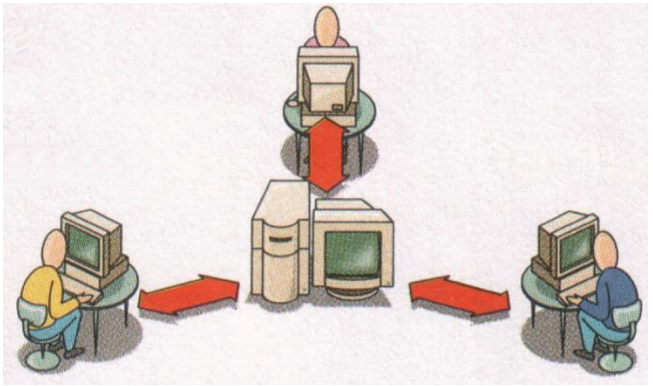
HyperText Transfer Protocol

- ✓ *HTTP : HyperText Transfert Protocole*
  - Un des protocole les plus courants sur Internet
    - Un protocole omniprésent: de IT à Embedded
  - Il est utilisé pour la navigation sur les sites Web
    - protocole de rapatriement des documents
    - protocole de soumission de formulaires

- ✓ Il en existe trois versions :
  - 0.9 (1991) : complètement obsolète
  - 1.0 (février 1997), de nos jours très rarement utilisée
  - 1.1 (octobre 2000). Les principaux changements entre les v1.0 et v1.1 sont l'ajout de 2 types de requêtes ainsi que la possibilité d'héberger plusieurs sites Web sur un même serveur dans la version 1.1.

Un Modèle Client - Serveur

- ✓ Modèle Client / Serveur
  - client: « browser » qui demande, reçoit et affiche des documents Web.
  - server: serveur Web qui envoie des documents en réponse aux requêtes des clients.



- ✓ L'échange entre le client et le serveur se fait en mode texte.
  - Le charset généralement utilisé est l'US-ASCII sur 8 bits.
  - Il est cependant possible que cet encodage soit modifié selon le client ou le serveur.

Principe de Fonctionnement de HTTP

- ✓ HTTP : TCP transport service
  - Le client initialise une connexion TCP/IP (voir sockets) sur le serveur et le port 80.
  - Le serveur accepte la connexion du client et fournit un port de communication (utilisateur).
  - Les messages http (messages au protocole de l'application) sont échangés entre le client http et le serveur http.
  - Enfin, la connexion TCP/IP est fermée.

- ✓ HTTP est "stateless"
  - En principe, le serveur ne maintient pas d'information sur les requêtes passées du client.
  - En pratique, certaines techniques le permette
  - HTTP : Transport ou Session ?

Zoom sur un Exemple de Communication

1. Le client http initialise une connexion TCP sur le serveur http *www.unice.fr*. (sur le port 80)
2. Le serveur http *www.unice.fr* en l'attente de connexions sur le port 80, accepte la demande de connexion du client
3. Le client http envoie un message de requête (contenant l'URL) au travers le socket de communication TCP.
4. Le serveur http reçoit le message de requête, compose le message de réponse contenant les objets demandés et renvoie le message au travers le socket de communication.
5. Le client http reçoit le message de réponse contenant le fichier HTML et l'affiche.
6. Le serveur http ferme la connexion.
7. En « parsant » le fichier HTML, le client http trouve 10 références à des objets jpeg. Les étapes 1 à 6 sont répétées pour chaque référence aux 10 objets jpeg.

Persistances des Connexions

- ✓ Connexions non-persistantes
  - HTTP/1.0
  - Le serveur « parse » la requête, répond et ferme la connexion.
  - Ralentit la récupération de la page complète
  - Mais la plupart des « browsers » 1.0 utilise des connexions parallèles
- ✓ Connexions persistantes
  - Par défaut pour HTTP/1.1
  - En-tête **Connection: Keep-Alive**
  - Durant une même connexion TCP, le serveur « parse » une requête, répond puis recommence ..
  - Le client envoie des requêtes pour tous les objets référencés aussitôt qu'il reçoit la page HTML de base.
  - Accélère la récupération de la page complète

En résumé

- ✓ Le fonctionnement de HTTP est très simple pour HTTP/1.0
  - connexion
  - demande (GET) d'un document
  - renvoi du document (status=200) ou d'une erreur
  - Déconnexion
- ✓ Cependant
  - Dialogue plus complexe en cas d'authentification
  - Optimisation : une série de plusieurs requêtes sur une connexion [Connexion « KeepAlive » de HTTP/1.1 (RFC 2068)]

Illustration sous Google Chrome

- ✓ Menu : Paramètres / Outils / Outils de développement
- ✓ Permet de tracer l'exécution des requêtes pour la récupération d'une page Web.

- ✓ De nombreuses manipulations sont alors possibles
  - Avec ou sans prétraitement
- ✓ Exemple :
  - Parsing de page et cascade de requêtes
  - Avec ou sans cache et temps de réponse
  - Quelques champs intéressants :
    - Pour www.polytech.unice.fr
    - Pour www.google.fr
  - Les cookies ? Où sont les cookies ?

Tests avec Telnet

- ✓ Telnet : un client TCP/IP Générique
- ✓ Permet d'établir une connexion TCP/IP avec n'importe quel serveur en mode caractère :
- ✓ telnet <serveur> <port>
- ✓ Exemple :
  - telnet www.polytech.unice.fr 80
  - Saisi de GET /
  - Retour ...

Dialogue HTTP

- ✓ Dialogue
  - en mode caractères ASCII (7 bits)
    - telnet www.sun.com 80
- ✓ Types de dialogue
  - Récupération d'un document
    - méthode GET
  - Soumission d'un formulaire
    - méthodes GET ou POST
  - Envoi de Document et Gestion de Site
    - méthodes PUT, DELETE, LINK, UNLINK
  - Gestion de proxy/cache
    - méthode HEAD (récupération des informations sur le document)

Structure

- ✓ **Les requêtes et les réponses sont bâties sur le même modèle**

```
{Ligne d'introduction}{SEP}
{En-têtes séparées par des {SEP}}
{SEP}{SEP}
{Corps}
```

- ✓ **Le seul élément capable de différencier une requête d'une réponse, c'est la** *Ligne d'introduction.*

Format de la Requête

<Méthode> <URI> HTTP/<Version>
[<Champ d 'entête>: <Valeur>]
[<tab><Suite Valeur si >1024>]
ligne blanche
[corps de la requête pour la méthode POST]

```
GET /docu2.html HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/gif
User-Agent:  Lynx/2.2  libwww/2.14
From:  alice@pays.merveilles.net
 * une ligne blanche *
```

```
POST /script HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/gif
User-Agent:  Lynx/2.2  libwww/2.14
From:  alice@pays.merveilles.net
Content-Length: 24
   * une ligne blanche *
name1=value1&
name2=value2
```

Source: Didier Donsez

Méthodes de la Requête

- ✓ GET
  - demande pour obtenir des informations et une zone de données concernant l 'URI
- ✓ HEAD
  - demande pour seulement obtenir des informations concernant l 'URI
- ✓ POST
  - envoie de données (contenu du formulaire vers le serveur, requête SOAP ...). Ces données sont situées après l'entête et un saut de ligne
- ✓ PUT
  - enregistrement du corps de la requête à l 'URI indiqué
- ✓ DELETE
  - suppression des données désignées par l 'URI

Méthodes de la Requête

- ✓ OPTIONS
  - demande des options de communication disponibles
- ✓ TRACE
  - retourne le corps de la requête intacte (débogage)
- ✓ LINK / UNLINK
  - association (et désassociations) des informations de l'entête au document sur le serveur
- ✓ Nouvelles extensions de WebDAV
  - PROPFIND, PROPPATCH, MKCOL,COPY, MOVE, LOCK, UNLOCK
- ✓ Nouvelles extensions HTTP/U HTTP/MU
  - NOTIFY, ... (UPnP)

Champs d'Entête

- ✓ Ils permettent la transmission d'informations complémentaires sur la requête, et le client lui-même.
- ✓ Ces champs agissent comme "modificateurs" de la requête, utilisant une sémantique identique à celle des paramètres passés par un appel d'une méthode de langage de programmation de haut niveau.

Format de la Réponse

HTTP/<Version> <Status> <Commentaire Status>
Content-Type: <Type MIME du contenu>
[< Champ d 'entête >: <Valeur>]
[<tab><Suite Valeur si >1024>]
Ligne blanche
Document

```
HTTP/1.0 200 OK
Date: Wed, 02Feb97 23:04:12 GMT
Server: NCSA/1.1
MIME-version: 1.0
Last-modified: Mon,15Nov96 23:33:16 GMT
Content-type: text/html
Content-length: 2345
    * une ligne blanche *
<HTML><HEAD><TITLE> ...
</BODY></HTML>
```

Source: Didier Donsez

Statuts des Réponses HTTP (RFC2068)

- ✓ 1xx Information
  - 100 : Continue (le client peut envoyer la suite de la requête), ...
- ✓ 2xx Succès de la requête client
  - 200: OK, 201: Created, 204 : No Content, ...
- ✓ 3xx Redirection de la Requête client
  - 301: Redirection, 302: Found, 304: Not Modified, 305 : Use Proxy,
- ✓ 4xx Requête client incomplète
  - 400: Bad Request, 401: Unauthorized, 403: Forbidden, 404: Not Found
- ✓ 5xx Erreur Serveur
  - 500: Server Error, 501: Not Implemented,
  - 502: Bad Gateway, 503: Out Of Resources (Service Unavailable)

Entêtes HTTP

Entêtes HTTP

- ✓ 4 types de champs d'entête
  - Général
    - Commun au serveur, au client ou à HTTP
  - Requête du client
    - formats de documents et paramètres pour le serveur
  - Réponse du serveur
    - informations concernant le serveur
  - Entité
    - informations concernant les données échangées

Entêtes Généraux

- ✓ Cache-Control
  - contrôle du caching.
- ✓ Connection = listes d'option
  - close pour terminer une connexion.
- ✓ Date
  - date actuelle (format RFC1123 mais aussi RFC850).
- ✓ MIME-Version
  - version MIME utilisé.
- ✓ Pragma
  - instruction pour le proxy.
- ✓ Transfer-Encoding
  - type de la transformation appliquée au corps du message.
- ✓ Via
  - utilisé par les proxys pour indiquer les machines et protocoles intermédiaires.
- ✓ ....

Entêtes de Requêtes Client (1)

- ✓ Accept
  - type MIME visualisable par l 'agent
- ✓ Accept-Encoding
  - méthodes de codage acceptées
  - compress, x-gzip, x-zip
- ✓ Accept-Charset
  - jeu de caractères préféré du client
- ✓ Accept-Language
  - liste de langues
  - fr, en, ...
- ✓ Authorization
  - type d 'autorisation
  - BASIC nom:mot de passe (en base64) (donc en transmis en clair!)
  - NB : Préalablement le serveur a répondu un WWW-Authenticate
- ✓ Cookie
  - cookie retourné

Entêtes de Requêtes Client (2)

- From
  - adresse email de l 'utilisateur
  - rarement envoyé pour conserver l'anonymat de l 'utilisateur
- Host
  - spécifie la machine et le port du serveur
  - un serveur peut héberger plusieurs serveurs
- If-Modified-Since
  - condition de retrait
  - la page n'est transférée que si elle a été modifiée depuis la date précisée. Utilisé par les caches
  - indique si le document demandé peut être caché ou pas.
- If-Unmodified-Since
  - condition de retrait
- ...

Entêtes de Requêtes Client (3)

- ✓ Max-Forwards
  - nombre max de proxy
- ✓ Proxy-Authorization
  - identification
- ✓ Range
  - zone du document à renvoyer
  - bytes=x-y (x=0 correspond au premier octet, y peut être omis pour spécifier jusqu 'à la fin)
- ✓ Referer
  - URL d 'origine
  - page contenant l'ancre à partir de laquelle le visualisateur a trouvé l'URL.
- ✓ User-Agent
  - modèle du visualisateur

Entêtes de Réponses Serveur

- ✓ Accept-Range
  - accepte ou refus d ′une requête par intervalle
- ✓ Age
  - ancienneté du document en secondes
- ✓ Proxy-Authenticate
  - système d ′authentification du proxy
- ✓ Public
  - liste de méthodes non standards gérées par le serveur
- ✓ Retry-After
  - date ou nombre de secondes pour un ressai en cas de code 503 (service unavailable)
- ✓ Server
  - modèle de HTTPD
  - utilisé par Satan !!!!
- ✓ Set-Cookie
  - crée ou modifie un cookie sur le client
- ✓ WWW-Authenticate
  - système d ′authentification pour l ′URI

Entêtes d'Entité (1)

- ✓ Allow
  - méthodes autorisées pour l 'URI
- ✓ Content-Base
  - URI de base
  - pour la résolution des URL
- ✓ Last-Modified
  - date de dernière modification du doc.
  - Utilisé par les caches
- ✓ Content-Length
  - taille du document en octet
  - utilisé par le client pour gauger la progression des chargements
- ✓ Content-Encoding
  - type encodage du document renvoyé
  - compress, x-gzip, x-zip
- ✓ Content-Language
  - le langage du document retourné
  - fr, en ...
- ✓ ...

Entêtes d'Entité (2)

- ✓ Content-MD5
  - résumé MD5 de l 'entité
- ✓ Content-Range
  - position du corps partiel dans l 'entité
  - bytes x-y/taille
- ✓ Content-Transfert-Encoding :
  - transformation appliqué du corps de l 'entité
  - 7bit, binary, base64, quoted-printable
- ✓ Content-Type
  - type MIME du document renvoyé
  - utilisé par le client pour sélectionner le visualisateur (plugin)
- ✓ Etag
  - transformation appliqué du corps de l'entité
  - 7bit, binary, base64, quoted-printable

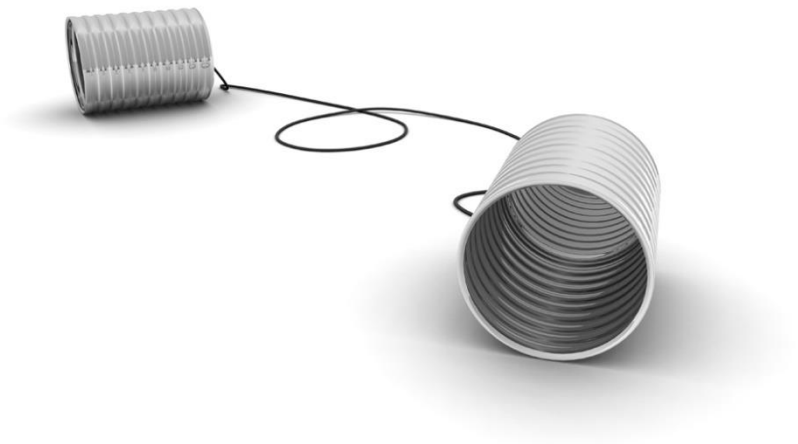
Entêtes d'Entité (3)

- ✓ Expires
  - date de péremption de l 'entité
- ✓ Last-Modified
  - date de la dernière modification de l 'entité
- ✓ Location
  - URI de l'entité
  - quand l 'URI est à plusieurs endroits
- ✓ URI
  - nouvelle position de l'entité
- ✓ ...

Internationalisation

- ✓ Langage Accepté
  - fr, de, it, en, sq (albanais), ru, (russe), ja (japonais),zh (chinois), el (grec), he (hébreu), ca (catalan) ...

- ✓ Charset (table de caractère)
  - par défaut ISO-8859-1 (Latin-1)
    - ISO-8859-2 (hongrois, albanais, ...)
    - ISO-8859- 4
    - ISO-8859-5, KOI8-R (russe, bulgare, polonais)
    - ISO-8859-7 (grec)
    - ISO-8859-8 (hébreu)
    - ISO-8859-9 (turc)
      Shift_JIS, ISO-2022-JP, EUC-JP (japonais
    - Big5 (chinois simplifié)
    - GB2312(chinois traditionnel - Taiwan)

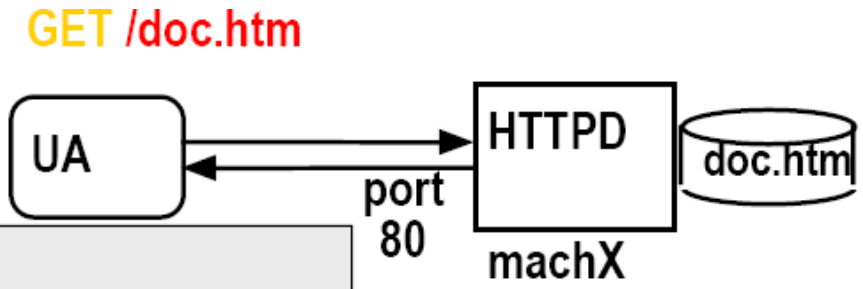


Echange de Documents

Réception et Envoi de Données

Récupération d'un Document Méthode GET

- ✓ GET /fichier



le Client envoie

GET /doc.htm HTTP/1.0 *méthode,chemin,version*
Accept: www/source *documents acceptés*
Accept: text/html
Accept: image/gif
User-Agent: Lynx/2.2 libwww/2.14
From: alice@pays.merveilles.net
* une ligne blanche *

le Serveur répond

HTTP/1.0 200 OK *ligne de status*
Date: Wed, 02Feb97 23:04:12 GMT
Server: NCSA/1.1
MIME-version: 1.0
Last-modified: Mon,15Nov96 23:33:16 GMT
Content-type: text/html *type du document retourné*
Content-length: 2345 *sa taille*
* une ligne blanche *
<HTML><HEAD><TITLE> ...

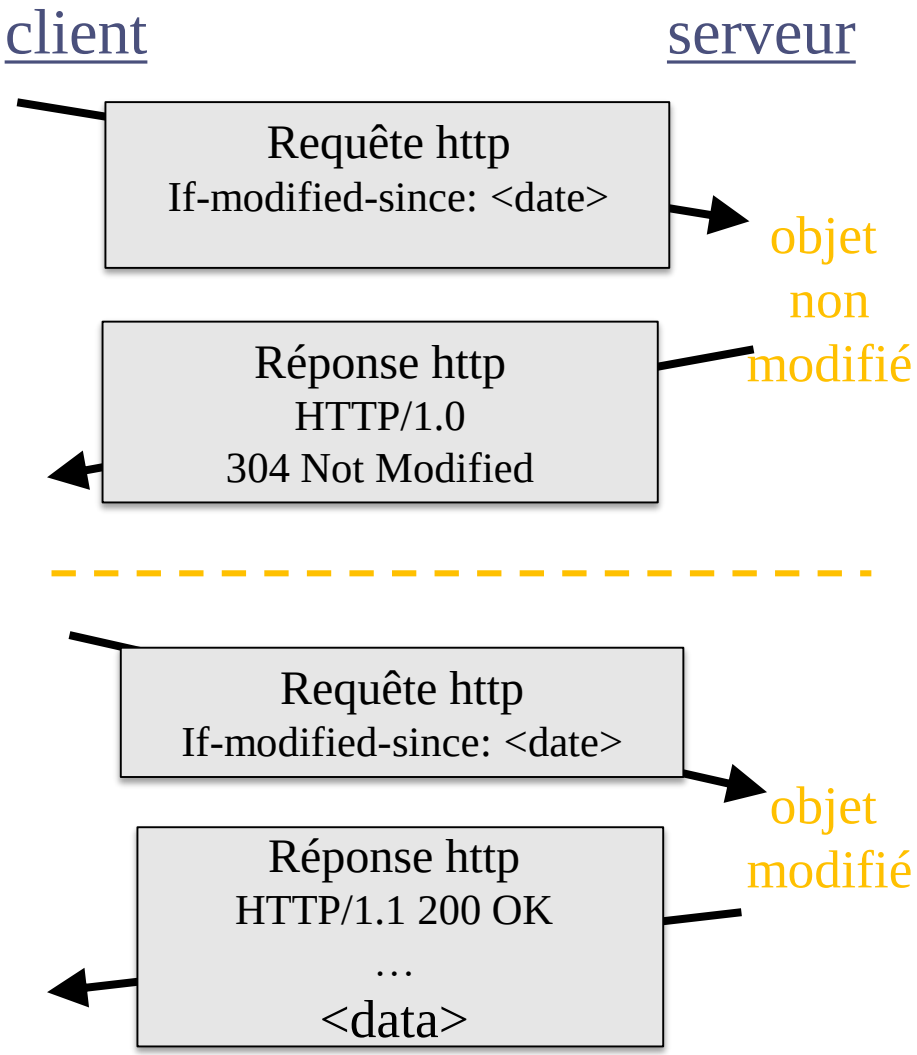
Récupération
Méthode GET conditionnelle

- ✓ **Objectif** : ne pas envoyer d'objet si le client à une version chargée à jour (en cache).
- ✓ **Client**: spécifie la date de la copie en cache dans la requête :

```
If-modified-since: <date>
```

- ✓ **Serveur**: la réponse ne contient pas de données si l'objet est à jour :

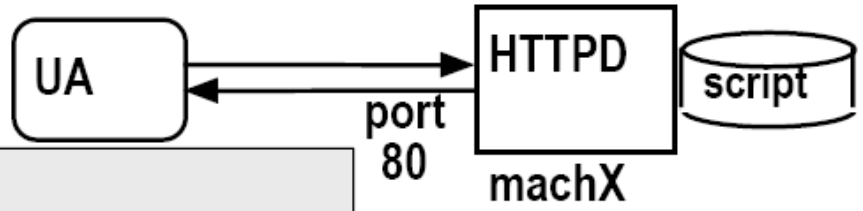
```
HTTP/1.0 304 Not Modified
```



Soumission d'un Formulaire Méthode GET

- ✓ GET/script?name1=value1&name2=value2

GET /script?name1=value1&name2=value2



le Client envoie

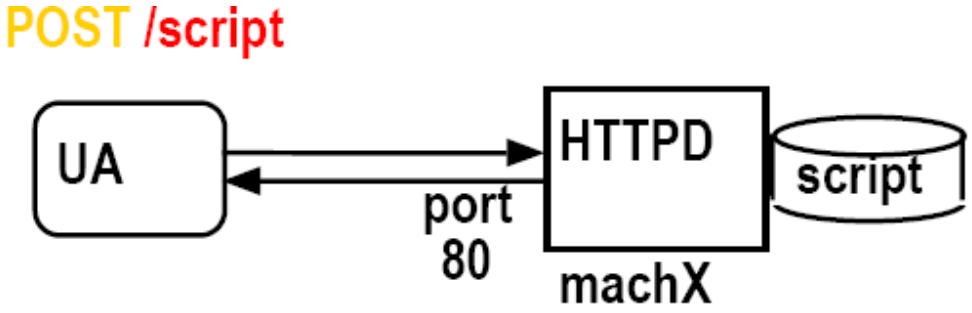
GET /script?name1=value1&name2=value2 HTTP/1.0
Accept: www/source *documents acceptés*
Accept: text/html
Accept: image/gif
User-Agent: Lynx/2.2 libwww/2.14
From: alice@pays.merveilles.net
* une ligne blanche *

le Serveur répond

HTTP/1.0 200 OK *ligne de status*
Date: Wed, 02Feb97 23:04:12 GMT
Server: NCSA/1.1
MIME-version: 1.0
Last-modified: Mon,15Nov96 23:33:16 GMT
Content-type: text/html *type du document retourné*
Content-length: 2345 *sa taille*
* une ligne blanche *
<HTML><HEAD><TITLE> ...

Soumission d 'un Formulaire méthode POST

- ✓ POST /script



le Client envoie

```
POST /script HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/gif
User-Agent:  Lynx/2.2  libwww/2.14
From:  alice@pays.merveilles.net
    * une ligne blanche *
name1=value1&
name2=value2
```

le Serveur répond

```
HTTP/1.0 200 OK
…
Content-length: 2345
    * une ligne blanche *
<HTML><HEAD><TITLE> ...
```

Codage des « paramètres »

- ✓ Les valeurs passées (URL et contenu des entrées des formulaires) doivent être sur 7 bits et sans caractères spéciaux

- ✓ Format d'encodage : x-www-form-urlencoded
  - Espace ⇒ « + »
  - Tous les caractères spéciaux et accentués ⇒ % code ascii
    - @ %40
    - é %e9
  - Les entrées des formulaires sont encodés dans une chaîne composée de paires (nom de l'entrée)=(valeur de l'entrée) séparé par des &

- ✓ nom=Dupont+Jean&adresse=3+rue+de+la+Gait%e9%0a75014+Paris

Comportement du Client / type du document retourné

- ✓ A partir du type MIME de Content-Type
  - Visualisation native
    - la fonction de visualisation est dans le noyau (core) du client
      - `text/html, image/jpeg`
  - Visualisation par plugin
    - la fonction est présente dans une DLL, un SO ou un JAR
    - elle est liée dynamiquement pour réaliser la visualisation
      - `world/vrml, text/tex`
  - Visualisation externe
    - la fonction n'est pas présente dans le client
    - le client rapporte le document et le sauvegarde dans un fichier temporaire
      - `video/mpeg, application/postscript`

Sessions avec HTTP

Comment réaliser le suivi de sessions ?

Suivi de Sessions avec HTTP (1)

- ✓ Motivations :
  - La notion de session est importante dans une application conversationnelle
    - commerce électronique
    - « j ´ajoute ce produit à mon panier (existant)»
  - Cependant HTTP est un protocole «stateless»
    - le serveur ne maintient pas d ´informations liées aux requêtes précédentes d ´un même client.
    - HTTP est donc « sessionless »
  - Comment implanter la notion de session sur plusieurs requêtes HTTP
    - documents, CGI, Servlet, ASP

Suivi de Sessions avec HTTP (2)

- ✓ Méthodes
  - Le serveur génère un identificateur de session et associe un état (et une date limite de validité) à une session
  - Le client renvoie l ´identificateur de session à chaque requête HTTP vers le serveur

- ✓ Echange et Stockage de l'identificateur de session
  - Input HIDDEN dans les formulaires
  - Réécriture des URLs EXTRA_PATH
  - Cookies (désactivable)
  - SSL (Secure Socket Layer) et https

Suivi de Sessions avec HTTP (3)

- ✓ Une session s'étend sur plusieurs requêtes
  - documents, CGI, SSS, Servlet, ASP
  - le serveur maintient un contexte de session et y associe un identifiant de session
- ✓ 3 solutions de suivi
  - input HIDDEN
    - contient l 'identifiant de la session
  - la Ré-écriture d 'URL
    - l 'identifiant dans chaque URL (dans les documents)
  - les Cookies
    - information positionnée par le serveur sur le client la durée de vie du cookie dépasse la session
    - puis envoyée par le client à chaque requête

Authentification avec HTTP

L'authentification dans HTTP

- ✓ Indiqué dans les ACL
- ✓ Modes d'authentification
  - BASIC
    - nom d 'utilisateur et mot de passe échangé en clair (base64) !
    - base des mots de passe dans un fichier htpasswd utilitaires de gestion du fichier
  - DIGEST
    - sécurisation de BASIC
    - hachage sécurisé MD5 du (nom,password,URI, méthode,nombre aléatoire fourni par le serveur)
  - SSL
    - Secure Socket Layer (TLS : Transport Layer Security)
    - authentification avec CA du serveur (2.0) et du client (3.0)
    - confidentialité avec DES
    - puis dialogue HTTP sur la connexion SSL

Contrôle d'Accès dans HTTP

- ✓ ACL (Access Control List)
  - spécifie les autorisations (ALLOW) ou les interdictions (DENY) d 'accès à une arborescence virtuelle du serveur
  - en fonction :
    - de l 'authentification
    - de la localisation du client sous domaine DNS, réseau ou adresse IP
- ✓ ACF (Access Control File)
  - fichier regroupant les ACL
    - global : access.conf dans Apache
    - par arborescence : .htaccess
  - combinaison des ALLOW et des DENY

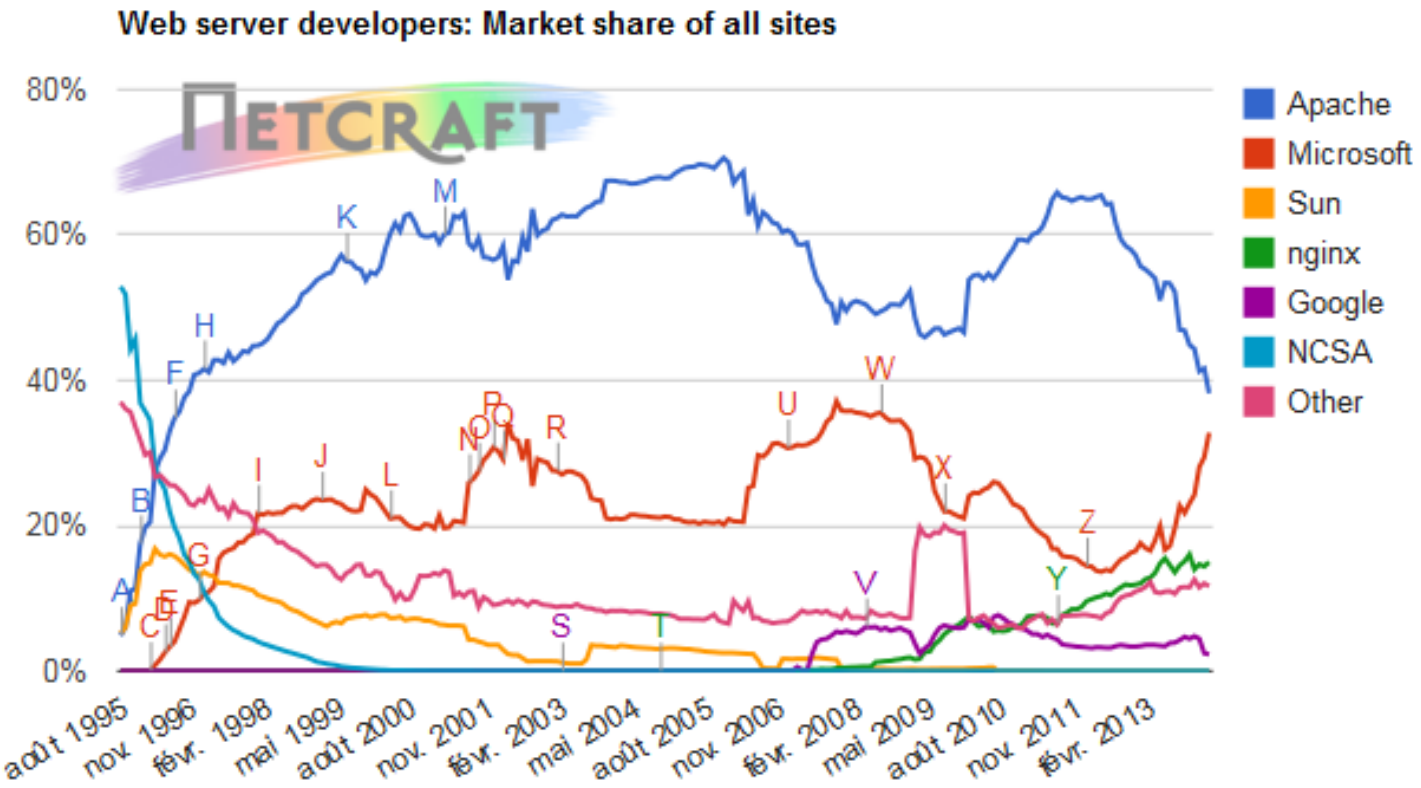
Audit des Requêtes

- ✓ Journaux des requêtes
  - les accès (access.log, refferee.log), et les erreurs (error.log),... sont journalisés
- ✓ Exploitation des Journaux
  - erreur dans les liens, ...
  - clientèle, analyse d'activité, ...
- ✓ Reporting (Présentation Synthétique)
  - Pour Apache
    - AccessWatch, Wusage,Analog, wwwstat
  - IIS, NS
    - intégré et visualisé par un script
  - Généraux
    - Net Analysis (Net Genesis), Enterprise Suite (Web Trends)

Infrastructure HTTP

Répartition part de Marché Serveurs Web

- ✓ Apache ou Microsoft ?



Source: http://news.netcraft.com/

Les Serveurs du Marché











- Apache Web Server
- Apache Tomcat
- Microsoft IIS Windows Web Server
- Nginx web server
- lighttpd web server
- The Jigsaw web server software from W3C
- Klone web server
- Abyss web server
- Oracle Web Tier
- X5 (formerly Xitami) web server
- Zeus Technology Ltd. - Zeus web server

Apache

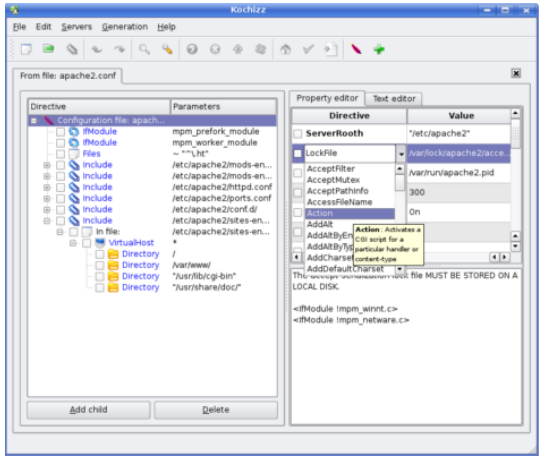
- ✓ A patch of NCSA HTTPD
  - serveur le plus répandu (« toujours » la ,plus grosse part de marché)
  - gratuit, issu du serveur NCSA HTTPD
  - très nombreuses plates-formes Unix et Windows NT
  - extensible par des modules tiers
- ✓ Nombreux Modules Tiers
  - possibilité d'étendre Apache avec des modules externe http://www.zyzzzyva.com/server/module_registry)
    - mod_auth_cookies_file, mod_auth_cookies_msql, mod_cgi_sugid, mod_perl, mod_perl_fast, mod_auth_kerb, mod_auth_dbi, mod_rewrite, mod_jserv(servlet), mod_java (CGI écrit en Java), php3
  - nombreux sous-projets autour de Java (Jakarta) et XML (Xerces, Xalan, XSP, Cocoon, ...)

Configuration Apache

- ✓ Fichiers de configuration
  - httpd.conf
    - comportement de base port TCP/IP, journaux, keepalive, UID, virtualhost, proxy, ...
  - Les autres fichiers sont rajouté à l'aide de l'instruction *Include*
    - Exemples :
    - Include /usr/local/apache2/conf/ssl.conf
      Include /usr/local/apache2/conf/vhosts/*.conf

Outil GUI : Kochizz, éditeur de configuration Apache.

…



2. Quelques Manipulations ....

✓ **1. Utilisation de Telnet pour contacter un serveur Web :**

```
telnet www.unice.fr 80
```

Ouvre une connexion sur le port 80 (port par défaut) de www.unice.fr

Tout ce qui est tapé est maintenant transmis au serveur sur le port 80

2. Envoi d'une requête GET

```
GET /index.html HTTP/1.0
```

En tapant ceci, vous envoyez cette requête GET, minimale mais complète au serveur http (suivi de 2 « retour chariot »).

3. Récupération de la réponse du serveur Web