Université Nice SOPHIA ANTIPOLIS

Le protocole HTTP

Les éléments du protocole

Structure des requêtes et des réponses

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

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

  - **{SEP} est un retour à la ligne**
  - **{SEP}{SEP}est donc une ligne vide**

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

Format des Requêtes

<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
```

<URI> contient ?name1=value1&name2=value2

Source: Didier Donsez

Méthodes des Requêtes

- ✓ **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 des Requêtes

- ✓ **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'Entêtes …

- ✓ Les Champs d'Entêtes sont en partie commun aux requêtes et aux réponses HTTP
- ✓ Nous les détaillerons plus loin dans le cours
- ✓ 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

Les 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 ...
- ✓ ...

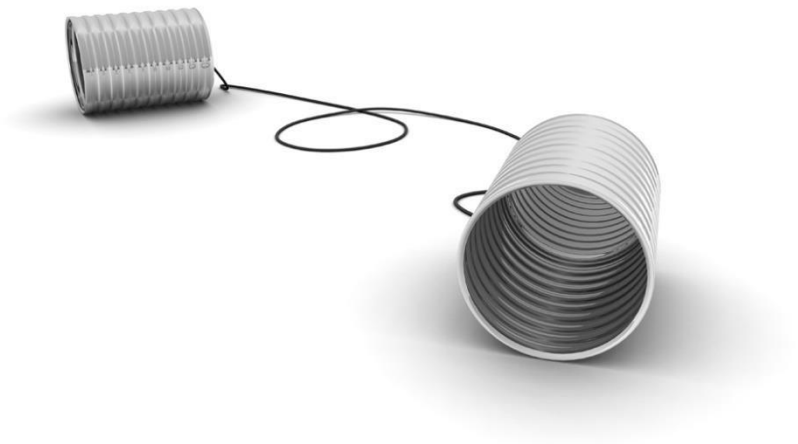
Entêtes d'Entité (2)

- ✓ **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
- ✓ **...**

Université Nice SOPHIA ANTIPOLIS

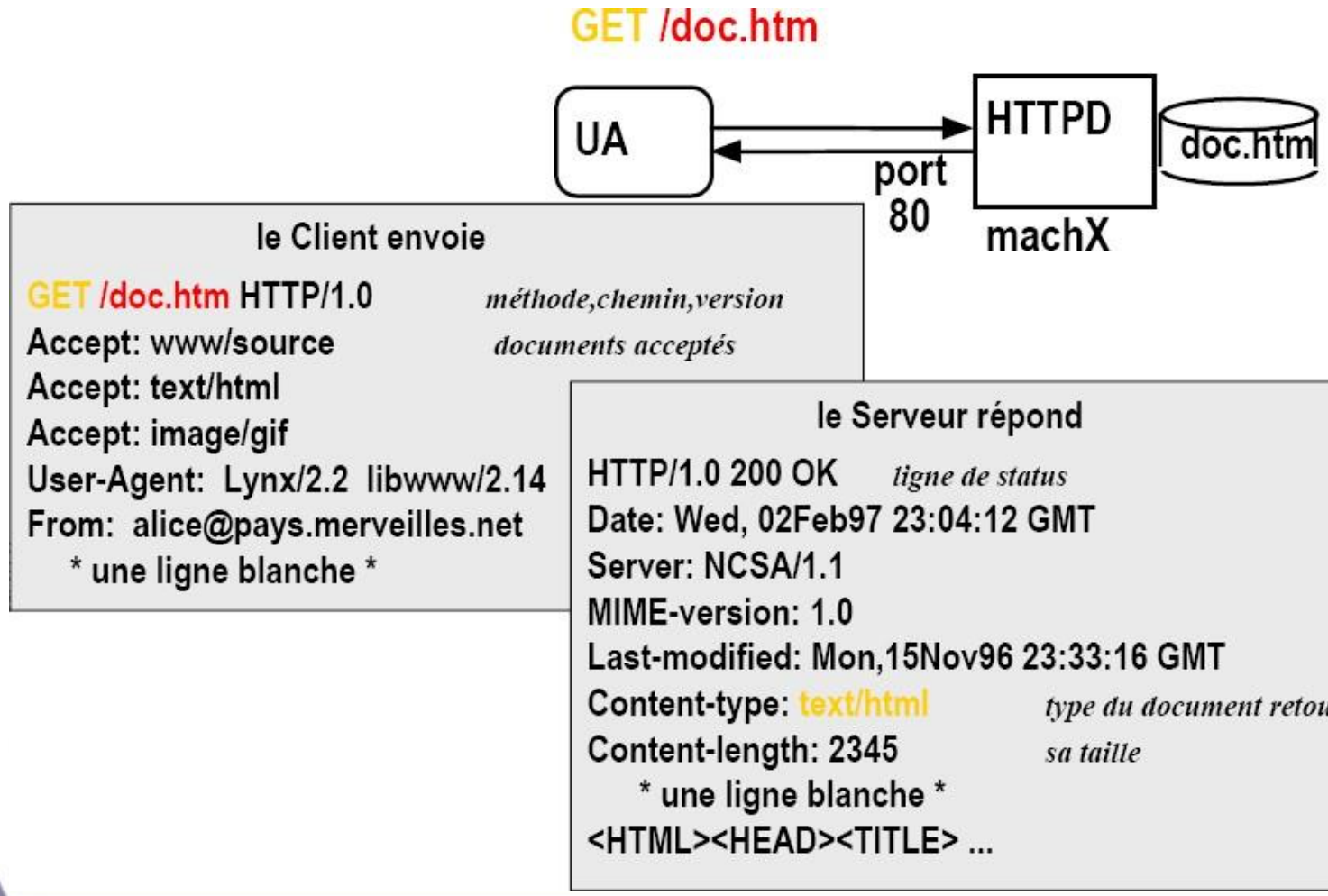


Pages Web Dynamiques

Réception et Envoi de Données

Récupération d'un Document Méthode GET

- ✓ GET /fichier

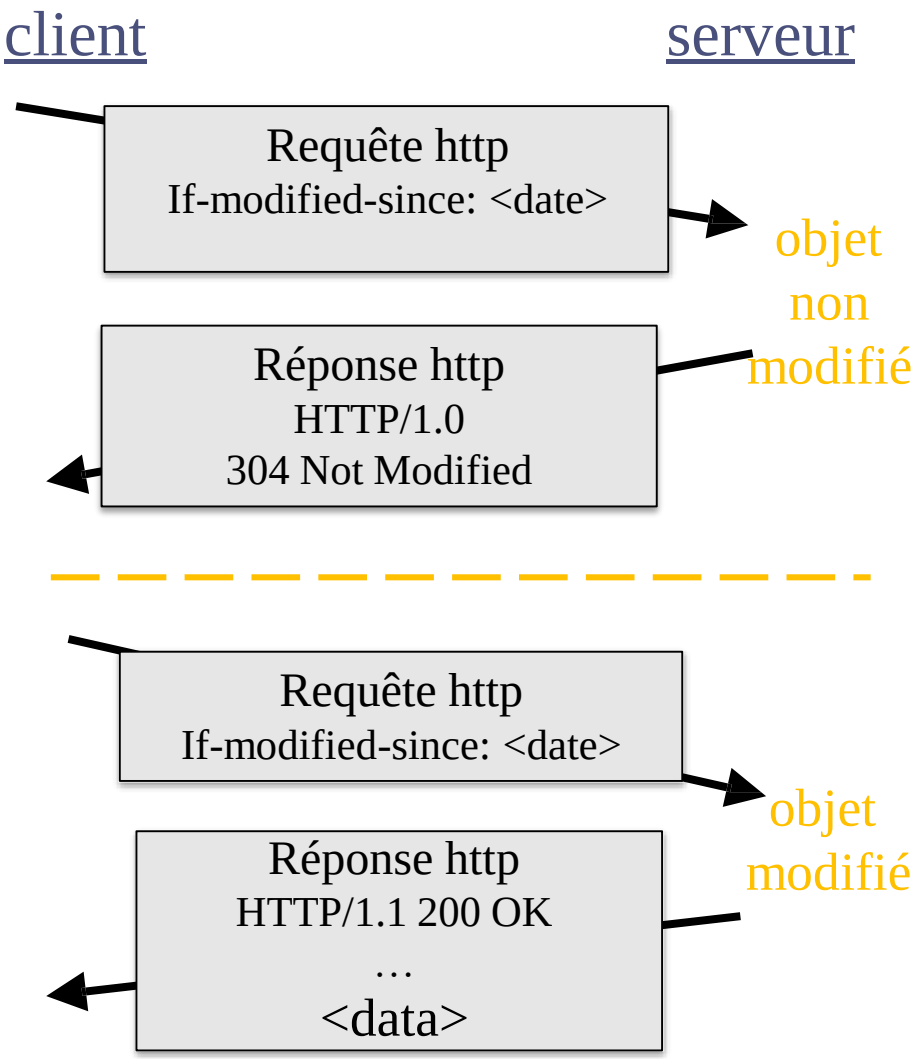


Récupération
Méthode GET conditionnelle

- ✓ **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 :**

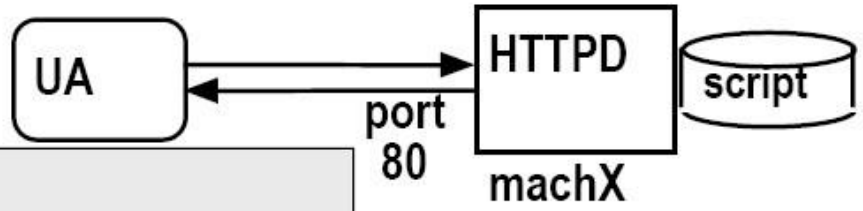
  `HTTP/1.0 304 Not Modified`



Soumission d'un Formulaire Méthode GET

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

GET /script?name1=value1&name2=value2



le Client envoie

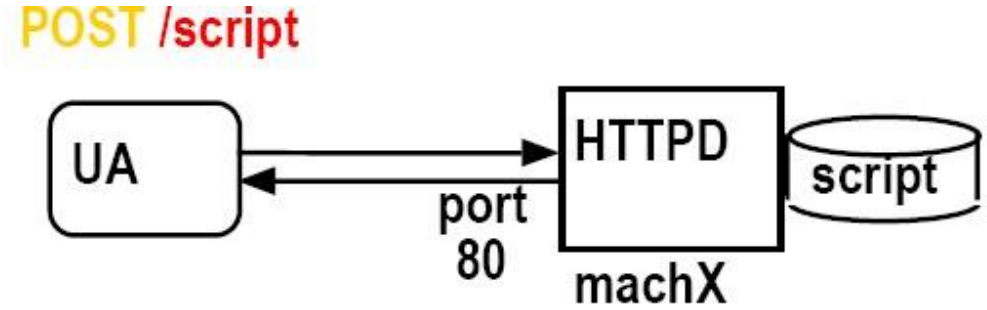
GET /script?name1=value1&name2=value2 HTTP/1.0
Accept: www/source *documents acceptés*
Accept: text/html
Accept: image/gif
User-Agent: Lynx/2.2 libwww/2.14
From: alice@pays.merveilles.net
* une ligne blanche *

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`