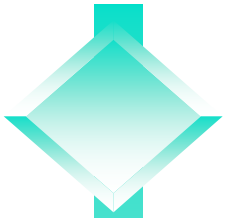
Cours n°8

Protocole HTTP (transfert hypertexte)



Sommaire

1. Dialogue HTTP

- Message HTTP
- Requête HTTP
- Réponse HTTP

2. Programmation réseau

- Client HTTP
- Serveur HTTP

Bibliographie

T. Berners-Lee, R. Fielding & H. Frystyk, Hypertext Transfer Protocol -- HTTP/1.0 , RFC 1945, 1996 [MIT Irvine]

R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach & **T. Berners-Lee, Hypertext Transfer Protocol -- HTTP/1.1, RFC 2616, 1999 [MIT Irvine W3C Compaq Xerox Microsoft]**

Agents HTTP

Deux agents (ou programmes conversationnels)

Communication inter-agents par l'intermédiaire de Sockets (TCP principalement)

User Agent (UA)

Navigateur (ou browser)

- Récupération d'un document puis visualisation de ce document

Aspirateur de sites (website copier)

- Recopie des pages et des documents d'un site web (miroir, cache web)

Meta-moteur de recherches (website copier)

- Requêtes sur un ensemble de moteur de recherche et collation des résultats

Server Agent (SA)

Transformation de l'URL en fichier ou en script (interprétation ou exécution)

Vérification d'identité

Envoi de la réponse au client

Mise à jour des journaux d'audit (log)

Principes

Etapes du dialogue client-serveur

1) Etablissement d'une connexion TCP
2) Envoi d'une requête par le User Agent au Server Agent
3) Envoi de la réponse par le Server Agent au User Agent
4) Si protocole HTTP/1.0 fermeture de la connexion TCP sinon aller en 2)

Ajout de nœuds intermédiaires avec l'utilisation de proxy, gateway, et tunnels
Utilisation d'un port quelconque (par défaut 80)

Transfert de fichiers

Envoi de fichiers par l'intermédiaire des requêtes ou des réponses,
codage texte (US-ASCII)

transcodage par les procédures MIME (non incluse dans http)

codage binaire

accélération des transferts mais moins efficace que le protocole ftp

Construction

Entête

Caractéristiques générales du message à transmettre (**`general-header`**)

Meta-données sur le corps du message (**`entity-header)`**

Description de la requête (**`Request-Line)`**

Informations complémentaires sur la requête (**`request-header)`**

Description de la réponse (**`Status-Line)`**

Informations complémentaires sur la réponse (**`response-header)`**

Corps du message **`[ message-body ]`**

Page HTML,

Tout type de ressource (avec codage et compression)

general-header

Caractéristiques générales du message à transmettre

```
general-header = Cache-Control | Connection | Date | Pragma
                 | Trailer | Transfer-Encoding | Upgrade | Via | Warning
```

Requête ou réponse HTTP
Pas de lien avec la ressource référencée par l'URI

Cache-Control Modalités des mécanismes de cache (choix de données à mémoriser, durée de conservation, politique de rechargement)

Connection Persistance de la connexion ou fermeture à la fin du dialogue

Date Horodatage du message (date et heure)

Transfer-Encoding Codage appliqué au corps du message (message-body)

Upgrade Demande de modification à chaud du protocole de communication

Warning Informations supplémentaires

entity-header

Meta-données sur le corps du message

```
entity-header  = Allow | Content-Encoding | Content-Language
               | Content-Length | Content-Location | Content-MD5
               | Content-Range | Content-Type | Expires | Last-Modified
```

Allow Liste des méthodes permises pour la ressource référencée par l'URI

Content-Type	Type de contenu
Content-Encoding	Type de compression utilisée
Content-Language	Langage prévue des destinataires de la ressource
Content-Length	Taille du corps du message
Content-Location	Localisation de la ressource
Content-MD5	Code de détection d'erreur (test d'intégrité)
Content-Range	Position du corps du message dans la ressource
Expires	Date d'expiration de la validité de la ressource
Last-Modified	Date de la dernière modification de la ressource

Format d'une requête

```
Request = Request-Line
          *(( general-header | request-header | entity-header ) CRLF)
          CRLF
          [ message-body ]

Request-Line   = Method SP Request-URI SP HTTP-Version CRLF

request-header = Accept | Accept-Charset | Accept-Encoding
               | Accept-Language | Authorization | Expect | From
               | Host | If-Match | If-Modified-Since | If-None-Match
               | If-Range | If-Unmodified-Since | Max-Forwards
               | Proxy-Authorization | Range | Referer | TE | User-Agent
```

Exemple de requête

```
HEAD /e-cursus/C2i/index.htm HTTP/1.1
Host: www.paris4.sorbonne.fr
Accept: www/source, text/html, image/gif
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.1;
Linux; fr)
From: ilgii1@laposte.net
Connection: Keep-Alive
Cache-control: no-cache
Accept-Encoding: x-gzip, x-deflate, gzip, identity
Accept-Charset: iso-8859-1, utf-8;
Accept-Language: fr, en
```

Request-Line

```
Request-Line = Method SP Request-URI SP HTTP-Version CRLF
```

Première ligne d'une requête HTTP

Trois champs

1) Type de requête

```
Method       = "OPTIONS" | "GET" | "HEAD" | "POST" | "PUT" | "DELETE"
             | "TRACE" | "CONNECT"
```

2) Adresse de la ressource

```
Request-URI  = "*" | absoluteURI | abs_path | authority
```

3) Version du protocole

```
HTTP-Version = "HTTP/0.9" | "HTTP/1.0" | "HTTP/1.1"
```

Types de requête

OPTIONS Demande d'information sur les options du serveur (capacité, types de requêtes traitées) et les caractéristiques des ressources hébergées (type de codage, langue, …)

HEAD Demande d'information concernant la ressource référencée par l'URI

GET Demande inconditionnelle ou conditionnelle de transfert d'une zone de donnée de la ressource référencée par l'URI (+ requête HEAD)

POST Envoi de données vers le serveur (en relation avec l'URI)
Annotation des ressources existantes, Postage d'un message (bulletin, liste de diffusion), Soumission d'un bloc de données à un processus de traitement de données (formulaire)

PUT Demande de stockage de données à l'URI indiqué (transfert montant d'un fichier)

DELETE suppression de la ressource référencée par l'URI

request-header (1/2)

Informations complémentaires sur la demande et le client

User-Agent Nom, version et caractéristiques du logiciel de User-Agent utilisé

Host Hôte Internet et port d'accès de la ressource demandée

Authorization Informations d'authentification pour la ressource demandée

Range Définition de la partie de la ressource référencée par l'URI à envoyer

Accept Types MIME acceptables pour la réponse (type + facteur q de proximité)

Accept-Charset Ensembles de caractères acceptables pour la réponse

Accept-Encoding Méthodes d'encodage acceptables pour la réponse (méthode + facteur q de proximité)

Accept-Language Langages acceptés pour la réponse (langage + facteur q de proximité)

request-header (2/2)

Exécution conditionnelle de la requête

If-None-Match V v différent de l'étiquette d'entité (code de version unique) de la ressource référencée par l'URI

If-Match V V égal à l'étiquette d'entité de la ressource référencée par l'URI

If-Modified-Since T T inférieur à la dernière date de modification de la ressource référencée par l'URI

If-Unmodified-Since T T supérieur à la dernière date de modification de la ressource référencée par l'URI

Tracabilité de la requête

From Adresse de messagerie du contrôleur du User Agent

Referer URI de la ressource contenant l'URI de la request-line (comportement de l'utilisateur)

Format d'une réponse

```
Response = Status-Line
           *(( general-header | response-header | entity-header ) CRLF)
           CRLF
           [ message-body ]

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF

response-header = Accept-Ranges | Age | ETag | Location | Proxy-
Authenticate | Retry-After | Server | Vary | WWW-Authenticate
```

Exemple de réponse

```
HTTP/1.1 200 OK
Date: Wed, 02 May 2007 21:32:13 GMT
Server: Apache/1.3.29 (Unix) PHP/4.4.1
Content-Length: 0
Allow: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS,
PATCH, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK,
UNLOCK, TRACE
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
```

Status-Line

Status-Line = **HTTP-Version** SP **Status-Code** SP **Reason-Phrase** CRLF

Première ligne d'une réponse HTTP

Trois champs

1) Version du protocole

HTTP-Version = "HTTP/0.9" | **"HTTP/1.0"** | **"HTTP/1.1"**

2) Code de la réponse

Status-Code = "100" | "101" | "200" | "201" | "202" | "203" | "204" | "205" | "206" | "300" | "301" | "302" | "303" | "304" | "305" | "307"| "400" | "401" | "402" | "403" | "404" | "405" | "406" | "407" | "408"| "409" | "410" | "411" | "412" | "413" | "414" | "415" | "416" | "417"| "500" | "501" | "502" | "503" | "504" | "505" |

3) Courte description textuelle du code de la réponse

Reason-Phrase = *<TEXT, excluding CR, LF>

Status-Code

1xx : Information

Demande reçue, poursuite du traitement
100 : Continue (le client peut envoyer la suite de la requête), ...

2xx : Succès

L'action a été bien reçue, comprise et acceptée
200: OK, 201: Created, 204 : No Content, ...

3xx : Redirection

Des actions ultérieures doivent être entreprises afin de mener la demande à bien
301: Redirection, 302: Found, 304: Not Modified, 305 : Use Proxy, ...

4xx : Erreur client

La demande contient une mauvaise syntaxe ou ne peut pas être satisfaite
400: Bad Request, 401: Unauthorized, 403: Forbidden, 404: Not Found

5xx : Erreur serveur

Le serveur a échoué à satisfaire une demande apparemment valide
500: Server Error, 501: Not Implemented,502: Bad Gateway, 503: Out Of Resources (Service Unavailable)

response-header

Informations complémentaires sur la réponse et le serveur

Server Nom, version et caractéristiques du logiciel de Server-Agent utilisé

ETag Etiquette d'entité (code de version unique) de la ressource référencée par l'URI

Accept-Ranges Acception ou refus d'une réponse partielle

Age Durée écoulée depuis la génération de la réponse

Location Redirection de la demande de l'utilisateur vers une autre URI

Proxy-Authenticate Demande d'authentification

Retry-After Temps d'indisponibilité du serveur

Requête GET (1/2)

RequeteGet.java

```
void Get (String chemin, String host, String fout) {
String sUA, sSA;
try {
FileWriter fw = new FileWriter(new File(fout));
sUA = "GET " + chemin + " HTTP/1.1\r\nHost: " + host + "\r\n";
System.err.print(sUA + " "); to.println(sUA);
// lecture de l'entête et recherche de la taille du corps
String tagTaille = "Content-Length: "; int taille = 0;
do {
      sSA = from.readLine(); System.err.println(sSA);
      if (sSA.startsWith(tagTaille) == true)
      taille = Integer.parseInt(sSA.substring(tagTaille.length()));
} while (sSA.length() != 0);
System.err.println("fin de l'entête");
System.err.println("la taille du corps du message est " + taille);
// lecture du corps et sauvegarde du fichier
int n = 0; int c;
do { c = from.read(); fw.write(c); n++; } while (n < taille);
fw.close();
} catch (Exception e) { System.err.println(e);}
}
```

Requête GET (2/2)

testRequeteGet.java

```
// Etablissement de la connexion
RequeteGet p = new RequeteGet("www.paris4.sorbonne.fr", 80);
// Envoi de la requête et lecture de la réponse
p.Get("/e-cursus/C2i/index.htm", "www.paris4.sorbonne.fr");
// fermeture de la connexion
p.fin();
```

```
Connexion établie entre /192.168.1.102:3096 et
www.paris4.sorbonne.fr/195.220.117.42:80
GET /e-cursus/C2i/index.htm HTTP/1.1
Host: www.paris4.sorbonne.fr
HTTP/1.1 200 OK
Date: Thu, 03 Apr 2007 16:55:03 GMT
Server: Apache/1.3.29 (Unix) PHP/4.4.1
Last-Modified: Wed, 02 Apr 2007 01:26:02 GMT
ETag: "a74e7-3594-4637e8aa"
Accept-Ranges: bytes
Content-Length: 13716
Content-Type: text/html
fin de l'entête
la taille du corps du message est 13716
```

ResponseGet.java

Réponse à la requête GET (1/3)

```
/** Reponse RGET
 * @param rep répertoire racine du site
 */
void RGet (String rep) {
String sUA, sSA;

try {
// lecture de l'entête et recherche de la ressource demandée
sUA = from.readLine(); System.err.println(sUA);
String[] tag = sUA.split(" ");
if ((tag[0].equals("GET")) == false) return;
File fin= new File(tag[1].replaceFirst("/", rep));
if (fin.exists() == false) {
sSA = " HTTP/1.1 404 NOT FOUND\n";
System.err.print(sSA + " "); to.println(sSA);
return;
}
do {
sUA = from.readLine(); System.err.println(sUA);
} while (sUA.length() != 0);
```

ResponseGet.java

Réponse à la requête GET (2/3)

```
// requête acceptée
sSA = " HTTP/1.1 200 OK";
System.err.println(sSA); to.println(sSA);
// taille du corps du message
FileReader fr = new FileReader(fin);
long taille = fin.length();
sSA = "Content-Length: " + taille +"\n";
System.err.println(sSA); to.println(sSA);
// lecture du fichier et creation du corps
int n = 0; int c;
do {
c = fr.read(); to.write(c); n++;
} while (n < taille);
fr.close();
to.println(); // fin du message
}
catch (Exception e) {
System.err.println(e);
}
}
```

testResponseGet.java

Réponse à la requête GET (3/3)

```
// Attente de la connexion
ReponseGet p = new ReponseGet(80);
// Réception de la requête et envoi de la réponse
p.RGet("cours09/");
// fermeture de la connexion
p.fin();
```

```
Connexion établie entre /192.168.1.104:80 et /192.168.1.102:1432
GET /index.htm HTTP/1.1
Host: 192.168.1.104
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.8.1.3)
Gecko/20070309 Firefox/2.0.0.3
Accept:text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,t
ext/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
 HTTP/1.1 200 OK
Content-Length: 13716
```