INF4375 - Paradigmes des échanges Internet

Le protocole HTTP

Jacques Berger

Objectifs

Présenter le protocole HTTP

Prérequis

Aucun

HTTP

HyperText Transfer Protocol

Version actuelle : 2

Protocole de couche applicative servant à échanger de l'hypertexte sur Internet

HTTP

HTTP est le protocole principal pour l'échange de données sur Internet

Normalisé par l'IETF et le W3C
- HTTP 1.1 en 1999

Fonctionnement

HTTP fonctionne selon un modèle client-serveur

Le serveur est un serveur web

Le client est un user agent
- Un fureteur
- Un fureteur vocal
- Tout dispositif accédant à du contenu web

Fonctionnement

Le user agent envoie une requête HTTP au serveur web

Le serveur web retourne une réponse HTTP au user agent

URL

L'URL est l'identifiant unique d'une ressource sur le web

Uniform Resource Locator

Il s'agit d'un cas particulier d'un URI (Uniform Resource Identifier)

URL

L'URL peut contenir :

- Le protocole de communication utilisé
- Le nom de domaine ou l'adresse IP
- Le port
- Une route
- Des paramètres

URL

Le protocole de communication utilisé est spécifié au tout début de l'URL

```
http://jberger.org/
```

Protocole utilisé : http

URL

Le nom de domaine est la partie principale de l'URL, c'est l'équivalent de l'adresse IP

```
http://jberger.org/
```

Nom de domaine : jberger.org

URL

Lorsque le port n'est pas précisé, HTTP utilise le port 80

Le port est précisé après le nom de domaine

```
http://jberger.org:3369/
```

Port : 3369

URL

La route est un chemin vers une ressource, après le nom de domaine

`http://jberger.org/articles/entretien-embauche`

Route : /articles/entretien-embauche

URL

Les paramètres sont des variables encodées dans l'URL

```
http://jberger.org/article?id=443&color=000
```

Paramètres : id=443 et color=000

Requête

Dans la requête HTTP, on spécifie :

- La méthode HTTP
- La version du protocole
- L'URL
- Les en-têtes HTTP
- Un body (si nécessaire)

Requête

Exemple

```
GET / HTTP/1.1
Host: jberger.org
Connection: keep-alive
Cache-Control: max-age=0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*
/*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.1916.114
Safari/537.36
Accept-Encoding: gzip,deflate,sdch
Accept-Language: fr-FR,fr;q=0.8,en-US;q=0.6,en;q=0.4
```

Réponse

La réponse contient :

- Le code de statut (résultat de la requête)
- Les en-têtes HTTP
- Un body (si nécessaire)

Réponse

Exemple

```
HTTP/1.1 200 OK
Date: Mon, 26 May 2014 02:32:55 GMT
Server: Apache
X-Powered-By: PHP/5.4.26
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html
```

Méthodes

La méthode détermine l'action à poser sur la ressource

Le protocole suggère une sémantique à chacune des méthodes HTTP

Le respect de cette sémantique n'est pas obligatoire mais elle est fortement recommandée

Méthodes

GET
Lecture de la représentation d'une ressource, ne devrait pas changer l'état de la ressource sur le serveur

HEAD
Lecture de l'en-tête de la ressource seulement, même chose que GET mais sans le body

Méthodes

POST
Création d'une ressource, envoi d'un formulaire

PUT
Remplacement d'une ressource

Méthodes

DELETE
Supprime une ressource

TRACE
Renvoie la requête reçue, l'équivalent d'un echo

Méthodes

OPTIONS
Retourne la liste des méthodes supportées pour une URL donnée

CONNECT
Conversion vers un autre protocole

Méthodes

PATCH
Appliquer des modifications partielles à une ressource

Méthodes

Les méthodes sécuritaires sont celles qu'on peut appeler sans crainte de modifier l'état de la ressource

Méthodes sécuritaires :

- GET
- HEAD
- OPTIONS
- TRACE

Méthodes

Les méthodes idempotentes sont celles que l'on peut invoquer plusieurs fois consécutives et que ça donne le même résultat que si la requête n'avait été faite qu'une fois

Méthodes idempotentes :

- PUT
- PATCH
- DELETE
- Méthodes sécuritaires

En-têtes

Les en-têtes HTTP sont des métadonnées sur l'information échangé

Elles ne sont pas destinées à l'utilisateur final, mais plutôt aux différents logiciels qui vont manipuler la requête et la réponse HTTP

En-têtes

L'en-tête la plus commune est :

Content-Type
Sert à définir le MIME type des données échangées

```
Content-Type:application/json
```

Body

Le body contient les données échangées

Les données peuvent être de plusieurs natures, par exemple :

- HTML
- JSON
- XML

Le format du body doit correspondre au Content-type

Body

Dans une requête, le body peut contenir plusieurs choses :

- Les données d'un formulaire
- Un document JSON qu'on veut envoyer au serveur
- Etc.

Body

Dans la réponse, le body contient les données qu'on veut envoyer au client :

- Document HTML
- Document XML
- Document JSON
- Etc.

Statuts

La réponse HTTP contient un code de statut sur 3 chiffres

Le code de statut permet de savoir si la requête a bien fonctionné

Statuts

Classes de statuts

1xx : Information
2xx : Succès
3xx : Redirection
4xx : Erreur côté client
5xx : Erreur côté serveur

Statuts communs

200 : OK
La requête a été traitée correctement. Réponse générale

201 : Created
Une nouvelle ressource a été créée

Statuts communs

301 : Moved permanently
L'adresse de la ressource a changé de façon permanente et une redirection doit être faite

400 : Bad request
La requête HTTP est invalide

Statuts communs

401 : Unauthorized
L'authentification a échoué ou elle n'a pas encore été fournie

403 : Forbidden
La requête est valide mais le serveur ne veut pas y répondre. L'authentification n'y change rien

Statuts communs

404 : Not found
La ressource demandée n'existe pas

500 : Internal Server Error
Erreur générique du serveur, le client ne peut rien faire

Statuts communs

501 : Not implemented
La fonctionnalité n'est pas disponible sur le serveur mais pourrait l'être éventuellement

503 : Service unavailable
Le service est présentement non disponible, état temporaire

Sécurité

La façon la plus commune d'encrypter un échange HTTP est d'utiliser HTTP Secure (https)

Plus loin...

RFC 2616 – HTTP/1.1
http://www.w3.org/Protocols/rfc2616/rfc2616.html

RFC 7540 – HTTP/2
https://tools.ietf.org/html/rfc7540