Programmation réseau et concurrente

Benoît Barbot

Département informatique, Université Paris-Est Créteil, M1

Mardi 8 mars 2016, Cours 5 : Web Service et Web API

Plan

Motivation

HTTP et Web Service

- HTTP conçu pour les échanges utilisateur-machine
- Web service échanges de donnée machine-machine
- Web service conçu par dessus HTTP
- Adapté au proxy

Motivation

HTTP et Web Service

- HTTP conçu pour les échanges utilisateur-machine
- Web service échanges de donnée machine-machine
- Web service conçu par dessus HTTP
- Adapté au proxy

Utilisation

- Échange entre serveurs
- Échange page web (JavaScript) <-> serveur
- Échange app mobile <-> serveur

Plan

Protocole HTTP

HyperText Transfer Protocol

- Construit par dessus TCP
- Textuelle
- Échange de données arbitraires (souvent textuelle)
- Basé sur la notion d' URL (Uniform Resource Locator)
- Sur le port 80 (Par convention)
- Standardisation des messages, pas des comportements
- Modèle client/serveur, le serveur n'initie jamais la connexion

Protocole HTTP

HyperText Transfer Protocol

- Construit par dessus TCP
- Textuelle
- Échange de données arbitraires (souvent textuelle)
- Basé sur la notion d' URL (Uniform Resource Locator)
- Sur le port 80 (Par convention)
- Standardisation des messages, pas des comportements
- Modèle client/serveur, le serveur n'initie jamais la connexion

Encodage URL

ASCII sans caractères spéciaux
Caractère spécial $\rightarrow$ "%code ascii"

Messages Client -> Serveur

Message

- {méthode} {URL} HTTP/1.1
- Host: {URL serveur}
- En-têtes
- Donnée (optionnelle)

Messages Client -> Serveur

Message

- {méthode} {URL} HTTP/1.1
- Host: {URL serveur}
- En-têtes
- Donnée (optionnelle)

Exemple navigateur web sur localhost

```
GET /test HTTP/1.1
Host: localhost:8080
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q
Accept-Language: fr-fr
Connection: keep-alive
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleW
```

Messages Serveur -> Clients

Message

- HTTP/1.1 {Code d erreur} {Description du code d erreur}
- En-tête
- Donnée (optionnelle)

Messages Serveur -> Clients

Message

- HTTP/1.1 {Code d erreur} {Description du code d erreur}
- En-tête
- Donnée (optionnelle)

Exemple sur www.example.com

```
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Type: text/html
Date: Thu, 03 Mar 2016 16:22:07 GMT
Etag: "359670651+gzip+ident"
Expires: Thu, 10 Mar 2016 16:22:07 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (iad/182A)
Vary: Accept-Encoding
X-Cache: HIT
x-ec-custom-error: 1
Content-Length: 1270

<!doctype html>
<html>...</html>
```

Méthodes

- GET demander un objet

Méthodes

- GET demander un objet
- POST ajouter des informations à un objet

Méthodes

- GET demander un objet
- POST ajouter des informations à un objet
- HEAD envoyer seulement les en-têtes

Méthodes

- GET demander un objet
- POST ajouter des informations à un objet
- HEAD envoyer seulement les en-têtes
- OPTIONS liste des types de messages disponibles

Méthodes

- GET demander un objet
- POST ajouter des informations à un objet
- HEAD envoyer seulement les en-têtes
- OPTIONS liste des types de messages disponibles
- *PUT* ajouter un objet
- *CONNECT* fabriquer un tunnel
- *DELETE* enlever un objet
- *TRACE* renvoie la requête
- *PATCH* applique un patch sur l'objet

En tête

Type de message

- Format {clé} : {valeur} {fin de ligne}
- Optionnel
- En théorie pas de limite de taille en pratique 100 lignes / 8Ko

En tête

Type de message

- Format {clé} : {valeur} {fin de ligne}
- Optionnel
- En théorie pas de limite de taille en pratique 100 lignes / 8Ko

En tête client $\rightarrow$ serveur

- “Cookie : Nom=Valeur ; ... “
- “User-Agent : ...“
- “Accept :” préférence type de contenue
- “Accept-Language : ” langue
- “Accept-Encoding : ” Format de compression
- “Connection : keep-alive” ne ferme pas la connections

En tête 2

En tête courante serveur → client

- "Set-Cookie : {Nom}={Valeur} ; Path=/ ;Expires={Date} ; Domain={.example.com} ; HttpOnly"
- "Server : Apache"
- "Connection : keep-alive" ne ferme pas la connections

Transmission de contenue

En un bloc

L'en-tête contient "Content-Length : Taille en octets"
suivi d'une ligne vide

Transmission de contenue

En un bloc

L'en-tête contient "Content-Length : Taille en octets"
suivi d'une ligne vide

En plusieurs blocs

- L'en-tête contient "Transfer-Encoding : chunked"
- Chaque bloc précédé par sa taille en octet
- fin du transfert par un bloc vide

Transmission de contenue

En un bloc

L'en-tête contient "Content-Length : Taille en octets"
suivi d'une ligne vide

En plusieurs blocs

- L'en-tête contient "Transfer-Encoding : chunked"
- Chaque bloc précédé par sa taille en octet
- fin du transfert par un bloc vide

En tête supplémentaire

- "Transfer-Encoding : compress/gzip"
- "Content-Type : text/html ;charset=ISO-8859-1"

Code d'erreur

1XX Requête reçu

- “100 Continue” en réponse à “Expect : 100-continue”
- “101 Switching Protocols” en réponse à “Upgrade”

Code d'erreur

1XX Requête reçu

- “100 Continue” en réponse à “Expect : 100-continue”
- “101 Switching Protocols” en réponse à “Upgrade”

2XX Succée

- “200 OK”
- “204 No Content”
- ...

Code d'erreur

1XX Requête reçu

- “100 Continue” en réponse à “Expect : 100-continue”
- “101 Switching Protocols” en réponse à “Upgrade”

2XX Succée

- “200 OK”
- “204 No Content”
- ...

3XX Redirection

- Contient un en-tête “Location :” avec une nouvelle URL
- “301 Moved Permanently”
- “302 Found”
- “303 See Other”
- ...

Code d'erreur 2

4XX Erreur Client

- “400 Bad Request”
- “403 Forbidden”
- “404 Not Found”
- ...

Code d'erreur 2

4XX Erreur Client

- “400 Bad Request”
- “403 Forbidden”
- “404 Not Found”
- ...

5XX Erreur Serveur

- 500 Internal Server Error
- ...

Méthodes Safe / Idempotent

Safe

Pas d'effet de bord

Méthodes Safe / Idempotent

Safe

Pas d'effet de bord

Idempotent

Deux appels successifs de la méthode équivalent à un seul appel

Méthodes Safe / Idempotent

Safe

Pas d'effet de bord

Idempotent

Deux appels successifs de la méthode équivalent à un seul appel

Méthodes

Méthodes	Safe	Idempotent
GET	✓	✓
POST	✗	✗
HEAD	✓	✓
OPTIONS	✓	✓
PUT	✗	✓
DELETE	✗	✓
PATCH	✗	✗

Proxy et Cache

Types

- Passerelle (gateaway)
- Transparent
- Proxy inverse, CND

Proxy et Cache

Types

- Passerelle (gateaway)
- Transparent
- Proxy inverse, CND

Cache

- Méthodes Safe peuvent être mis en cache
- Test de l'âge d'un objet
- Header : "Cache-Control : max-age=age en s, public/privée, no-cache "
- Header : "Date : Mon, 07 Mar 2016 09 :58 :44 GMT"
- Header : "Last-Modified : Mon, 07 Mar 2016 09 :56 :41 GMT"

Authentification

- Serveur demande une autorisation : 401 Unauthorized
  + dans l'en-tête un champ "WWW-Authenticate"
- Client ajoute dans l'en-tête un champ
  "Authorization :"

Authentification

- Serveur demande une autorisation : 401 Unauthorized
  + dans l'en-tête un champ "WWW-Authenticate"
- Client ajoute dans l'en-tête un champ
  "Authorization :"
- Authentification simple : "WWW-Authenticate : Basic
  realm="Message" "
  "Authorization : Basic {user :mdp en base64}"

HTTPS

- HTTP sécurisé
- Ouverture d'une connexion TLS/SSL
- Une fois la connexion ouverte HTTP standard

Utilisation de l'API Java

HttpURLConnection

- Gère les connexions HTTP
- Créé à l'aide de URL
- Setter et getter pour gérer les en-têtes et méthodes
- Input/Output stream pour le contenu

Utilisation de l'API Java

HttpURLConnection

- Gère les connexions HTTP
- Créé à l'aide de URL
- Setter et getter pour gérer les en-têtes et méthodes
- Input/Output stream pour le contenu

Demo

HttpURLConnection

Utilisation de l'API Java

HttpURLConnection

- Gère les connexions HTTP
- Créé à l'aide de URL
- Setter et getter pour gérer les en-têtes et méthodes
- Input/Output stream pour le contenu

Demo

HttpURLConnection

Demo

HttpsURLConnection

Plan

Web API

Principe

- Utilisation HTTP
- Objet envoyé sont des données sérialisées
- Souvent XML/JSON
- Les clients ne sont pas des navigateurs web

Démo météo

```
http://openweathermap.org/data/2.5/weather?appid=
44db6a862fba0b067b1930da0d769e98&q=Creteil&mode=xml
```

API Rest (Restfull)

Propriétés

- Client-serveur
- Sans état pour le serveur
- Utilisation des Caches
- Système hiérarchique
  - Ressource identifiée par URI
  - Message indépendant
  - Message contient des liens vers d'autres ressources
- Interface uniforme

Web service - SOAP

Simple Object Access Protocol

- Implémente une couche objet entre des machines
- Objet référencé par des URL
- Méthode référencée par des balises XML
- Utilisé pour les applications d'entreprise distribué

Soap - Exemple

Exemple

POST /InStock HTTP/1.1
Host : www.example.org
Content-Type : application/soap+xml ; charset=utf-8
Content-Length : 299
SOAPAction : "http ://www.w3.org/2003/05/soap-envelope"

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope'
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <m:GetStockPrice xmlns:m="http://www.example.org/stock/Surya">
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

Soap - Exemple

Exemple

POST /InStock HTTP/1.1
Host : www.example.org
Content-Type : application/soap+xml ; charset=utf-8
Content-Length : 299
SOAPAction : "http ://www.w3.org/2003/05/soap-envelope"

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope'
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <m:GetStockPrice xmlns:m="http://www.example.org/stock/Surya">
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

Demo

Eclipse web service