

Web et échange de données

Avènement du web

les entreprises ont dut fournir à leur client et partenaire de plus en plus de services « à la demande » depuis les années 2000.

- Du fait de l'augmentation des capacité réseau et infrastructure.
- Du fait de la prolifération des « providers Internet » qui ont aussi contribué à la démocratisation des technologies dites « web ».

HTTP

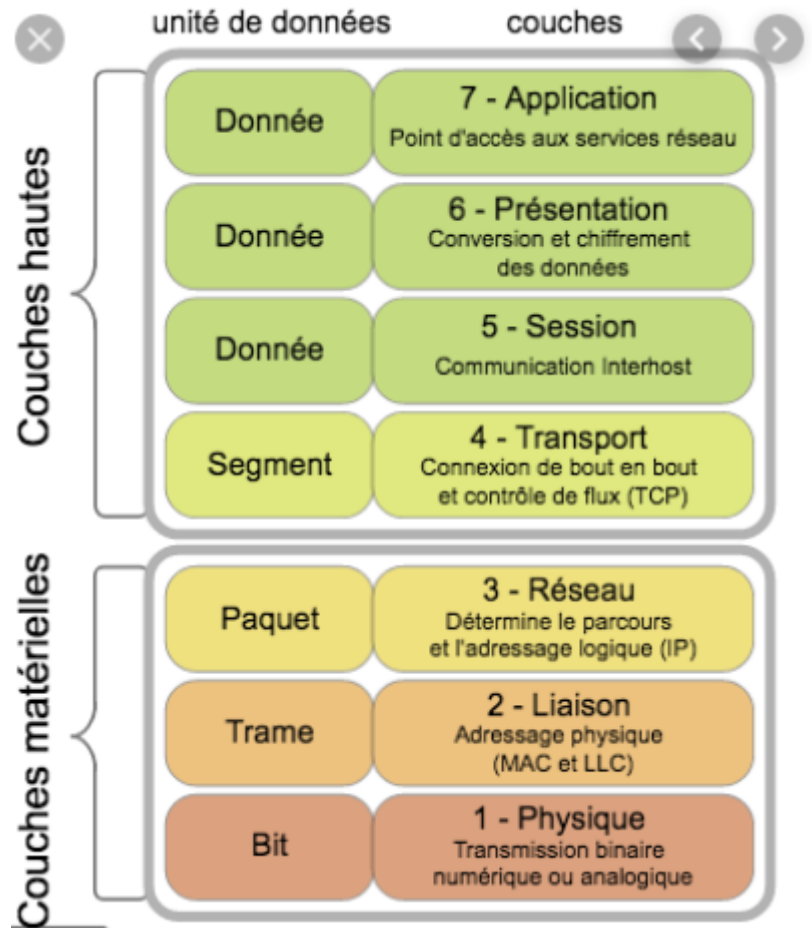
Définition

Ainsi le protocole HTTP s'est imposé comme fondement « réseau » de la plupart des échanges entre application.

L'HyperText Transfer Protocol, plus connu sous l'abréviation HTTP — littéralement « protocole de transfert hypertexte » — est un protocole de communication client-serveur développé pour le World Wide Web.

HTTPS (avec S pour secured, soit « sécurisé ») est la variante du HTTP sécurisée par l'usage des protocoles SSL ou TLS.

HTTP est un protocole de la couche application. Il peut fonctionner sur n'importe quelle connexion fiable, dans les faits on utilise le protocole TCP comme couche de transport.



La version actuelle de http est la version 1.1. Elle a été publiée par IETF en Février 2014 dans les RFC 7230 à 7237.

HTTP est un protocole client-serveur à commande : c'est-à-dire qu'il demande à un serveur de faire quelques choses et de renvoyer quelques choses à l'appelant.

La commande est encapsulée dans une requête les éventuels retour dans une réponse.

Le protocole HTTP/2 diffère de HTTP/1.1 sur plusieurs aspects:

- Il est encodé en binaire plutôt qu'en texte. Il ne peut donc plus être lu ou écrit à la main. Malgré cette difficulté, il est désormais possible d'implémenter des techniques d'optimisation avancée.
- C'est un protocole multiplexé. Plusieurs requêtes en parallèle peuvent être gérées au sein de la même connexion, supprimant ainsi la limitation séquentielle de HTTP/1.x.

HTTP/2 compresse les en-têtes, étant donné que des en-têtes similaires sont échangés lors d'une suite de requêtes, on supprime ainsi la duplication et l'échange inutiles des données similaires.

Il permet au serveur de remplir le cache du client avant qu'il ne soit demandé par ce dernier, on parle alors d'évènements générés par le serveur.

Fonctionnement

La requête est composée d'une zone d'entête et d'un corps.

Les entêtes sont codifiées mais extensibles

Le corps référence une URI et une méthode à exécuter.

La réponse est composée d'une zone d'entête et d'un corps.

Les entêtes sont codifiées mais extensibles

Le corps comporte ou pas du contenu et peut être facultatif selon la méthode utilisée.

Méthodes

GET est la méthode la plus courante pour demander une ressource. Une requête GET est sans effet sur la ressource, il doit être possible de répéter la requête sans effet.

HEAD ne fait que demander des informations sur la ressource, sans demander la ressource elle-même.

POST transmet des données en vue d'un traitement à une ressource. L'URI fourni est l'URI d'une ressource à laquelle s'appliqueront les données envoyées.

OPTIONS obtient les options de communication d'une ressource ou du serveur en général.

CONNECT pour utiliser un proxy comme un tunnel de communication.

TRACE demande au serveur de retourner ce qu'il a reçu, dans le but de tester et effectuer un diagnostic sur la connexion.

PUT remplace ou ajoute une ressource sur le serveur. L'URI fourni est celui de la ressource en question.

PATCH fait une modifie partiellement une ressource, contrairement à PUT.

DELETE supprime une ressource du serveur.

Données

Problématique

Informations et documents dans l'entreprise

Constat :

- Inflation du volume d'informations
- Coût et difficulté :
- Difficultés à retrouver un document
- Coût induit par la gestion de documents

Pour structurer cette « masse » d'information, il faut la transformer en donnée.

Une donnée est une information structurée.

Cela peut se faire soit en travaillant directement sur les informations ou en en ajoutant d'autres afin de pouvoir plus facilement retrouver ce que l'on recherche.

W3C

Le WORLD WIDE WEB Consortium crée des standards pour le Web.

Le W3C en 7 points...

- Accès universel (internationalisation, web mobile...)
- Web sémantique (RDF, XML...)
- Confiance (signatures électroniques, collaboratif...)
- Intéropérabilité (consensus industriel)
- Evolutivité (simplicité, modularité, compatibilité...)
- Décentralisation (systèmes répartis, tolérance aux erreurs)
- Multimédia (SVG, SMIL, HTML5)

Pour plus d'informations : www.w3c.org

Principales normalisations

| | 1986 : SGML | 1991 : HTML | 1998 : XML |
|----------------------|------------------------|------------------|--------------------|
| Objectifs | adaptabilité | simple | puissance de SGML |
| | intelligence | portable | Simplicité de SGML |
| | gestion des liens | gestion de liens | |
| Inconvénients | complexe | non adaptable | |
| | difficilement portable | non intelligent | |

XML

Définition

eXtensible Markup Language

- Recommandation (norme) du W3C
- Spécifiant un langage
- Constitué d'un ensemble d'éléments appelés balises
- Utilisable pour créer d'autres langages

2 concepts fondamentaux

- Structure, contenu et présentation sont **séparés**
- Les balises ne sont pas **figées**

Conséquences :

- XML est un format de document
- XML est un format de données

- XML est un mode de structuration de l'information
- XML est un méta-langage

Caractéristiques

Richesse sémantique de SGML

- Compatible SGML
- Dédié au traitement des données
- Soutient une grande variété d'applications
- Assure un entretien aisé
- Intelligent

Facilité de mise en œuvre de HTML

- Simple et lisible
- Portable et facilement utilisable sur Internet
- Assure un développement aisé

Exemple

```
<song>
<title> Ma chanson </ title >
<composer> par l'auteur</ composer >
< producer > Dupond</producer>
<editor> Maison edition</ editor >
<duration > 6:20</duration>
<date> 1978</ date >
< artist > Toto</artist>
</song>
```

- Balises "propriétaires" compréhensibles à la profession, dérivation des langages propriétaires.
- XML sépare le contenu de son aspect (à la différence de HTML ou tout peut être mélangé)

Structure

entete

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="presentation1.xsl"?>
<exposition>
```

Contenu

```
<tableau>
  <titre>Déjeuner sur l'herbe</titre>
  <artiste>Edouard Manet</artiste>
</tableau>
```

```

<tableau>
  <titre>Déjeuner des Canotiers</titre>
  <artiste>Auguste Renoir</artiste>
</tableau>
<tableau>
  <titre>Waterloo Bridge, le soleil dans le brouillard</titre>
  <artiste>Claude Monet</artiste>
</tableau>
<tableau>
  <titre>Mardi Gras, soleil couchant, bd Montmartre</titre>
  <artiste>Camille Pissaro</artiste>
</tableau>
<tableau>
  <titre>Les Joueurs de cartes</titre>
  <artiste>Paul Cézanne</artiste>
</tableau>
</exposition>

</xml>

```

SPECIFICATIONS

Résumé des spécifications :

- Un document doit commencer par une déclaration XML
- Toutes les balises avec un contenu doivent être fermées
- Toutes les balises sans contenu doivent se terminer par les caractères />
- Le document doit contenir un et un seul élément racine
- Les balises ne doivent pas se chevaucher
- Les valeurs d'attributs doivent être entre guillemets
- La casse doit être respectée pour toutes les occurrences de noms de balise (MAJUSCULES ou minuscules).

Un document respectant ces critères est dit "bien formé"

Arbre XML

Exemple de fichiers XML :

```

<adresses>
  <!--Carnet d'adresses-->
  <personne id= "personnel">
    <type classe= "famille"/>
    <nom>Durand</nom>
    <prenom>Martine</prenom>
    <adr:adresse>6, rue des Magnolias</adr:adresse>
  </personne>
</adresses>

```