

# Communication sur le Web

*Olivier Coupelon*

*2021-2022*

---

## Besoin

- Systèmes d'informations hétérogènes
    - Windows / MAC / Unix ...
    - Répartis aux quatre coins du monde
    - Yahoo, Flickr, Youtube...
- Communication entre ces services
- 

## Communication

- Communication Homme/Machine
    - Interface de communication visuelle
    - Affichage et saisie d'informations
  - Communication Machine/Machine
    - Interface de communication numérique
    - Envoi et réception d'informations entre systèmes hétérogènes
- 

## World Wide Web (WWW)

Système hypertexte public fonctionnant sur Internet

---

## URI

Uniform Resource Identifier

- Uniform Resource Locator :
  - <http://www.google.fr>
  - <mailto:coupelon@isima.fr?subject=Bonjour>
- Uniform Resource Name
  - [doi:10.1340/309registries](https://doi.org/10.1340/309registries)

- [http://fr.wikipedia.org/wiki/Schéma\\_d'URI](http://fr.wikipedia.org/wiki/Schéma_d'URI)

**http://server:port/path/to/resource?p1=v1&p2=v2**

---

## HTTP

HyperText Transfer Protocol

- Echange **synchrone** de documents (Request / Response)
  - **Le** protocole d'échange sur le Web
  - **Un** protocole d'échange sur Internet
  - Actuellement en v1.1 massivement, v2.0 disponible.
- 

## HTTP/2

- Basé sur SPDY (Google)
  - RFC 7540 publiée en mai 2015
  - Novembre 2015 : 2,3% des 10 millions de sites les plus consultés le supporte
  - Novembre 2018 : 31% des 10 millions de sites les plus consultés le supporte
  - Décembre 2020 : 50% des 10 millions de sites les plus consultés le supporte
  - Possibilité de transmission en binaire, donc moins lisible, mais plus performant
- 

## HTTP/3

- Basé sur QuIC (Google), Quick UDP Internet Connection
  - Spécification de janvier 2020 : <https://quicwg.org/base-drafts/draft-ietf-quic-http.html>
  - TCP n'a pas évolué depuis 1970
  - Remplacement de **TCP** par QuIC pour optimisation des temps de connexion et de la bande passante
  - Fin 2020, 5% des navigateurs le supportent, aucun sur mobile
  - Fin 2020, 4% du top 10 millions de sites le supportent
- 
-



Figure 1: HTTP/3

## Requête HTTP

```
GET /index.html HTTP/1.1
Host: www.example.com
```

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Accept-Ranges: bytes
Content-Length: 10
Connection: close
Content-Type: text/html; charset=UTF-8
```

```
<html><body>Hello world!</body></html>
```

---

## Méthodes HTTP

- **GET** : demande la représentation d'une ressource.
  - **HEAD** : idem, mais sans le contenu (horodatage)
  - **POST** : envoi de données pour traitement
  - **PUT** : mise à jour d'une ressource
  - **PATCH** : mise à jour d'une partie de ressource
  - **DELETE** : suppression d'une ressource
  - **OPTIONS** : liste les méthodes supportées par la ressource
  - TRACE, CONNECT
  - Les navigateurs web utilisent GET, POST et parfois HEAD.
- 

## Codes retour HTTP

- Indique au client comment la requête a été traitée et que faire de la réponse donnée.
  - 200=ok, 201=created
  - 301=moved, 307=temporary redirect
  - 401=unauthorized, 404=not found
  - 500=internal error, 501=not implemented
- Code à la base du web, doivent **toujours être respectés**.

---

## Type MIME

Multipurpose Internet Mail Extensions

- Décrivent le type de données échangées Content-Type: text/plain
  - *text/plain, text/html, text/xml, text/css, text/javascript, video/mpeg, video/mp4, image/gif, image/png, image/jpeg, audio/mpeg, audio/x-wav, application/pdf, application/json, application/octet-stream, application/zip, (...)*
- 

## Sécurité du protocole

- Déclinaison TLS (SSL) du HTTP : HTTPS
    - Transport Layer Security (Secure Sockets Layer)
  - Authentification du protocole :
    - Basic : Login/Mot de passe transmis en clair sur le réseau.
    - Digest : Mot de passe codé, plus sécurisé.
- 

## HTTP : un protocole sans état

- Stateless : deux requêtes consécutives sont sans lien pour HTTP.
  - Mécaniques de sessions :
    - Coté serveur : HTTPSession, qui pour un identifiant de session donné stocke des informations.
    - Coté client : cookie qui contient l'identifiant de session, et d'autres informations conservées côté client. Ces informations ne peuvent être critiques, car modifiables par le client.
- 

## Construction de la réponse HTTP

- Contenu statique (images, pages statiques, fichiers) :
  - L'accès à la ressource est fait en fonction de la requête HTTP demandée, le contenu est simplement copié
- Contenu dynamique :
  - Les données sont calculées en fonction des paramètres en entrée et renvoyées au client.

- C’est le serveur Web qui est configuré pour déterminer la nature du contenu.
- 

## Common Gateway Interface (CGI)

- Protocole normalisé pour invoquer un programme externe (Perl, C++, Python, Ruby, ...)
  - Des variables d’environnement sont positionnées (IP client, chemin de requête, etc) puis le programme est invoqué.
  - Les flux standard sont utilisés (réponse, traitement POST, ...).
  - Cette solution est flexible mais a un coût certain : chaque requête entraîne la création / destruction d’un processus.
- 

## FastCGI

- ... même principe que CGI, à la différence que les processus restent en mémoire.
  - 1 processus peut traiter plusieurs requêtes (multiplexing et connexions multiples).
  - Possibilité de lancer plusieurs processus “en cluster”.
  - Utile dans de nombreux cas pour Ruby on Rails, Django, etc.
- 

## Embarqué dans le serveur Web

- Les programmes sont embarqués dans le serveur web :
    - mod\_python
    - mod\_perl
    - mod\_php
    - Servlets
    - (...)
  - Approche relativement simple, mais :
    - attention aux crashes / sécurité
    - alourdit le serveur
    - attention à la montée en charge
-

## Conteneur pour un serveur d'application

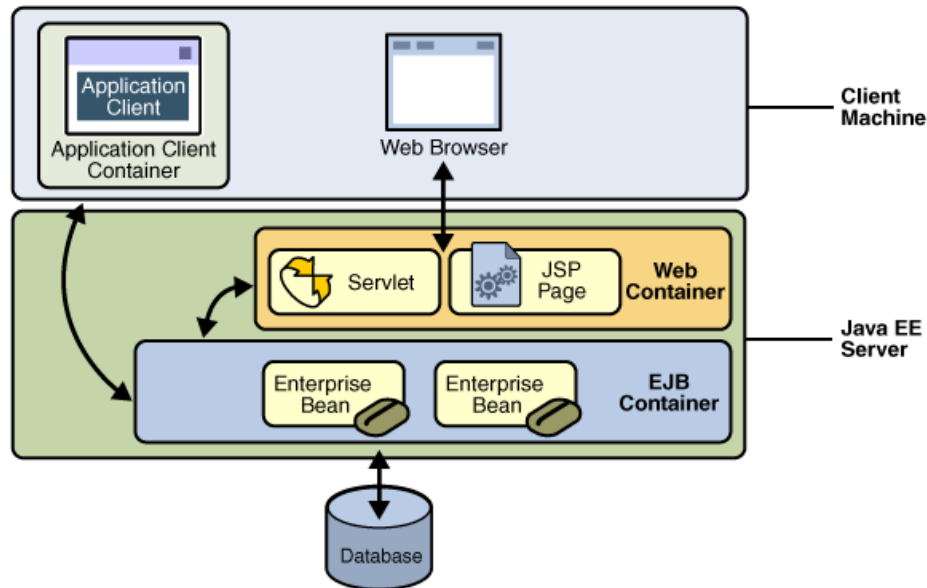


Figure 2: Conteneurs

---

## Conteneur Web

- Conteneur de servlet
  - Tomcat est le plus connu.
  - Glassfish, WebSphere, JBoss... embarquent un conteneur web.
- Les servlets permettent le paramétrage et la gestion de :
  - Mapping d'URL
  - Filtres
  - Authentification
  - (...)
- Surtout utilisé pour l'**exposition de web services**, les pages web sont faites par du développement front appelant ces derniers.

---

## Conteneur web en Java (sooooo 2000)

```
import java.io.*;
```

```
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/plain");
        response.getWriter().write("Hello World");
    }
}
```

HelloWorld.java

---

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema"
    <display-name>HelloWorld Application</display-name>
    <description>
        This is a simple web application with a source code organization
        based on the recommendations of the Application Developer's Guide.
    </description>
    <servlet>
        <servlet-name>HelloServlet</servlet-name>
        <servlet-class>examples.Hello</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>HelloServlet</servlet-name>
        <url-pattern>/hello</url-pattern>
    </servlet-mapping>
</web-app>
```

web.xml