

# Support de cours Réseaux

Dr Yani-Athmane BENNAI

---

Université de Béjaia

Faculté des Sciences Exactes

Département d'Informatique

24 décembre 2024

# Table des matières

<b>3</b>	<b>Aperçu de la couche application</b>	<b>3</b>
3.1	L'architecture du Web . . . . .	3
3.1.1	Type de service : Client-serveur distribué . . . . .	3
3.1.2	Exemple : Récupération d'une page Web simple avec images intégrées . . . . .	3
3.1.3	Exemple : Récupération d'une page web avec des liens externes . . . . .	4
3.2	Hypertexte et hypermédia . . . . .	4
3.3	Navigateurs Web . . . . .	5
3.3.1	Fonction principale . . . . .	5
3.3.2	Architecture des navigateurs . . . . .	5
3.4	Le serveur Web . . . . .	6
3.5	URL (Uniform Resource Locator) . . . . .	6
3.6	Types de documents Web . . . . .	6
3.6.1	Documents statiques . . . . .	6
3.6.2	Documents dynamiques . . . . .	7
3.6.3	Documents actifs . . . . .	8
3.7	HTTP : HyperText Transfer Protocol . . . . .	8
3.7.1	Fonctionnement . . . . .	8
	HTTP utilise TCP . . . . .	8
	HTTP est « sans état » (Stateless) . . . . .	9
3.7.2	Connexions HTTP : 2 types . . . . .	9
3.7.3	Messages HTTP . . . . .	10
	Message de requête HTTP . . . . .	11
	Message de Réponse HTTP . . . . .	12
3.8	DNS : Domain Name System . . . . .	13
3.8.1	DNS est une base de donnée hiérarchique . . . . .	13
3.8.2	Les étapes générales d'une résolution DNS . . . . .	14

---

# Table des figures

3.1	Récupération d'une page web avec des liens externes . . . . .	4
3.2	Le principe d'Hypertexte . . . . .	5
3.3	Architecture générale d'un navigateur . . . . .	6
3.4	Obtention d'un document dynamique . . . . .	7
3.5	Exemple d'une chaîne de requête . . . . .	8
3.6	Applets Java . . . . .	8
3.7	Intéraction Client/Serveur dans HTTP . . . . .	9
3.8	Connexions HTTP . . . . .	9
3.9	HTTP non-persistant . . . . .	10
3.10	HTTP persistant . . . . .	10
3.11	Format d'une requête HTTP . . . . .	11
3.12	Quelques méthodes HTTP . . . . .	11
3.13	Quelques Headers de requête HTTP . . . . .	12
3.14	Format d'une réponse HTTP . . . . .	13
3.15	Quelques Headers de réponse HTTP . . . . .	13
3.16	Hiérarchie des serveurs DNS . . . . .	14

## Chapitre 3

# Aperçu de la couche application

### Introduction

La couche application, située au sommet du modèle TCP/IP, agit comme une interface directe entre les utilisateurs et le réseau. Elle fournit des services réseau adaptés aux besoins des applications des utilisateurs, tels que la navigation sur le Web, l'envoi de courriels ou encore le partage de fichiers.

### 3.1 L'architecture du Web

L'architecture du Web repose sur un modèle de service client-serveur distribué, conçu pour permettre aux utilisateurs d'accéder facilement à des ressources en ligne.

#### 3.1.1 Type de service : Client-serveur distribué

Dans cette architecture, le client, généralement un navigateur Web, agit comme une interface permettant à l'utilisateur de soumettre des requêtes pour des ressources ou des services hébergés sur un serveur. Le serveur, de son côté, répond à ces requêtes en fournissant les ressources demandées, comme des pages Web ou des fichiers.

Les sites Web, qui regroupent les pages Web, servent d'emplacements hébergeant ces ressources. Ces pages Web peuvent être :

- Simples, ne contenant pas de liens hypertextes vers d'autres ressources.
- Composites, avec des liens hypertextes qui permettent de naviguer entre plusieurs documents, enrichissant ainsi l'expérience utilisateur.

#### 3.1.2 Exemple : Récupération d'une page Web simple avec images intégrées

Lorsqu'un utilisateur demande une page Web contenant des images intégrées (stockées dans le même fichier), le processus est simple :

1. Le client envoie une seule requête au serveur pour obtenir la page complète.
2. Le serveur transmet en une seule réponse l'ensemble du contenu, y compris les images et le texte.

Cette approche est efficace lorsque toutes les ressources nécessaires sont contenues dans un fichier unique.

### 3.1.3 Exemple : Récupération d'une page web avec des liens externes

Les pages composites, en revanche, peuvent contenir des références à des ressources externes (images ou documents hébergés ailleurs). Le processus de récupération suit ces étapes :

1. Le client envoie une requête pour le document principal (fichier A).
2. Le navigateur identifie les ressources externes référencées dans le fichier principal, comme une image (fichier B) ou un texte (fichier C).
3. Des requêtes supplémentaires sont alors envoyées aux serveurs concernés pour récupérer ces ressources.

Ce mécanisme illustre la capacité des navigateurs à assembler dynamiquement des contenus dispersés.

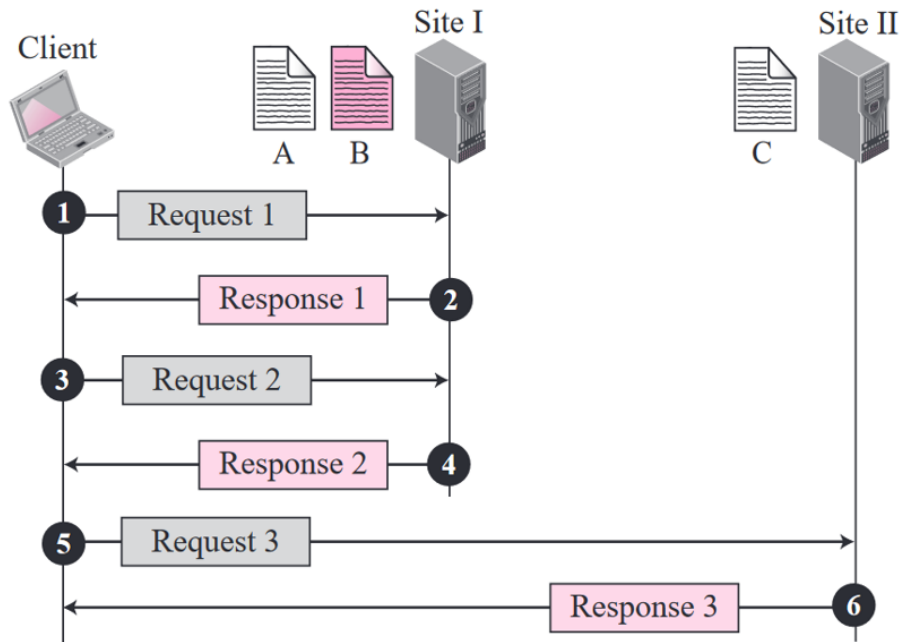


FIGURE 3.1 – Récupération d'une page web avec des liens externes

## 3.2 Hypertexte et hypermédia

L'hypertexte et l'hypermédia sont des éléments fondamentaux de la navigation sur le Web.

- Hypertexte : Il s'agit de documents textuels incluant des liens vers d'autres documents. Ces liens, cliquables dans un navigateur, permettent une navigation rapide entre les contenus.

- **Hypermédia** : Ce concept élargit celui de l'hypertexte en intégrant des médias supplémentaires, tels que des images, des vidéos ou des fichiers audio, directement accessibles via des liens.

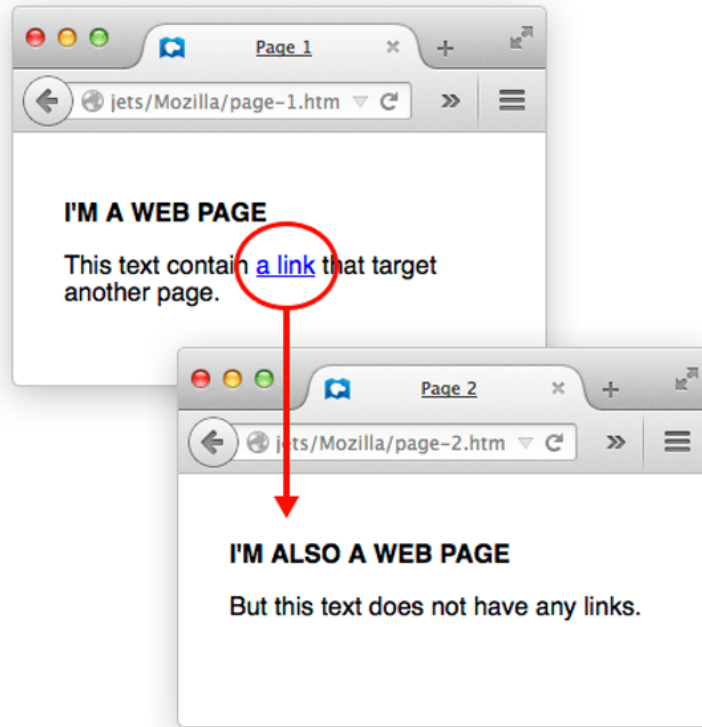


FIGURE 3.2 – Le principe d’Hypertexte

## 3.3 Navigateurs Web

### 3.3.1 Fonction principale

Un navigateur Web est un logiciel permettant d’accéder aux pages Web. Il interprète les documents reçus des serveurs et les affiche de manière compréhensible pour l’utilisateur. Ce rôle crucial repose sur l’utilisation de plusieurs outils et protocoles.

### 3.3.2 Architecture des navigateurs

L’architecture d’un navigateur comprend trois composants principaux :

- Le contrôleur : Gère les interactions avec l’utilisateur via le clavier ou la souris et pilote l’affichage des documents.
- Les protocoles clients : Ils assurent la communication avec les serveurs (FTP, HTTP, etc.).
- Les interpréteurs : Chaque interpréteur est spécialisé dans un type de contenu :

**HTML** pour les pages Web standard.

**Java** pour exécuter des applets Java.

**JavaScript** pour gérer des scripts interactifs et dynamiques.

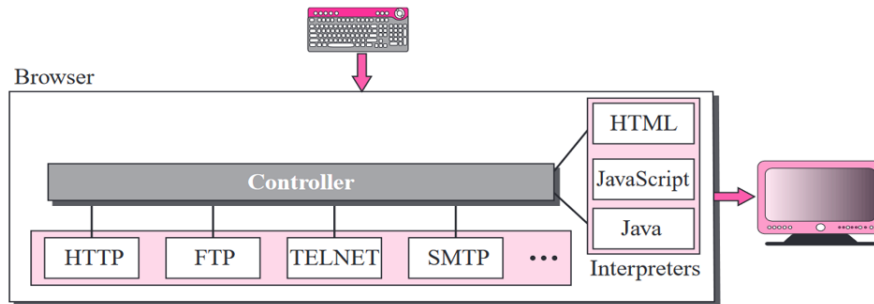


FIGURE 3.3 – Architecture générale d'un navigateur

## 3.4 Le serveur Web

Un serveur Web est responsable de l'hébergement et de la distribution des pages Web. Voici son fonctionnement :

1. Les pages sont stockées sur le serveur et servies aux clients à la demande.
2. Un système de cache optimise les performances en conservant temporairement les fichiers fréquemment demandés en mémoire.
3. Grâce au multiprocesseur, le serveur peut traiter plusieurs requêtes simultanément, garantissant ainsi une meilleure performance.

## 3.5 URL (Uniform Resource Locator)

L'URL est l'adresse unique permettant de localiser une ressource sur Internet. Une URL typique se compose de :

- Protocole : Spécifie le mode de communication (ex. HTTP, HTTPS).
- Hôte : Indique le serveur/nom de domaine (ex. www.exemple.com).
- Port : Facultatif, par défaut 80 pour HTTP.
- Chemin : Précise l'emplacement exact de la ressource sur le serveur.

## 3.6 Types de documents Web

### 3.6.1 Documents statiques

Les documents statiques ont un contenu fixe défini lors de leur création. Ces fichiers ne peuvent être modifiés que par leur auteur sur le serveur. Exemples typiques : des pages HTML simples ou des fichiers PDF. Leur simplicité les rend idéaux pour des contenus à faible interactivité.

### 3.6.2 Documents dynamiques

Contrairement aux documents statiques, les documents dynamiques sont générés à la demande par le serveur, souvent à l'aide de scripts ou de programmes. Par exemple, une page affichant l'heure actuelle est créée dynamiquement à chaque requête, car l'heure change constamment.

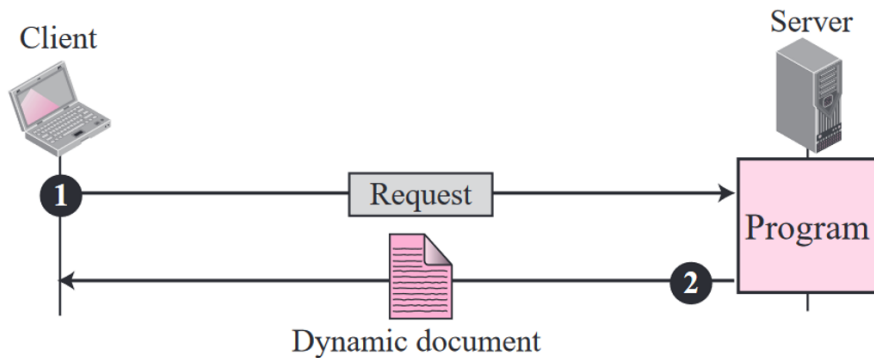


FIGURE 3.4 – Obtention d'un document dynamique

Un aspect important de cette interaction dynamique est l'utilisation des Query Strings (ou chaînes de requête). Ces dernières permettent au client de transmettre des paramètres au serveur directement via l'URL.

Ce mécanisme peut être comparé à l'entrée traditionnelle d'un programme, où des paramètres sont spécifiés lors de l'exécution, comme dans le cas d'un programme de copie où les noms des dossiers source et destination sont passés en arguments.

Dans le cadre de la communication navigateur-serveur, la query string agit comme un appel de fonction :

- Si les données transmises sont simples et de petite taille (par exemple, un mot ou une valeur), elles peuvent être directement ajoutées à l'URL.
- Prenons l'exemple suivant : **http://www.deanza/cgi-bin/prog.pl?23**. Ici, la valeur "23" est passée au serveur sous forme de query string.

Lors de la réception de cette requête :

1. Le serveur analyse l'URL et la sépare au niveau du point d'interrogation "?".
2. La partie avant le "?" identifie le programme ou script à exécuter (prog.pl dans cet exemple).
3. La partie après le "?" est interprétée comme les données fournies par le client ("23" dans cet exemple), qui sont ensuite stockées dans une variable utilisable par le programme.

Ce principe est utile pour créer des interactions dynamiques, comme des formulaires en ligne ou des filtres de recherche, où les paramètres spécifiés par l'utilisateur influencent directement le contenu retourné par le serveur.



https://www.domain.com/url?variable=value&variable=value

separator

start of query string

FIGURE 3.5 – Exemple d’une chaîne de requête

### 3.6.3 Documents actifs

Les documents actifs exécutent des scripts directement sur le navigateur du client. Ces fonctionnalités permettent des interfaces interactives et réactives. Exemples :

- Applets Java : Applications complexes exécutées dans le navigateur.
- JavaScript : Scripts plus légers et faciles à intégrer pour des pages dynamiques.

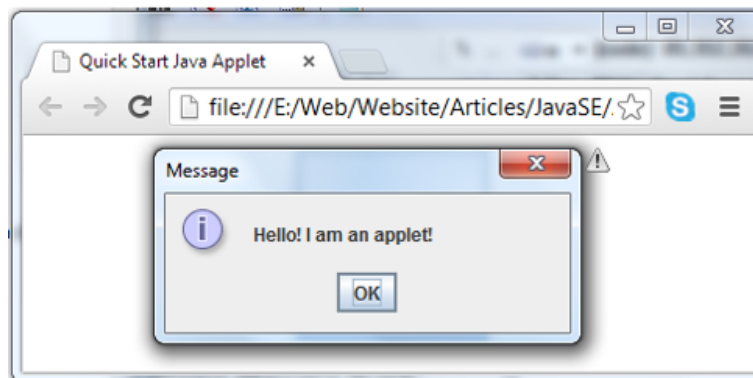


FIGURE 3.6 – Applets Java

Différences avec Java :

- JavaScript est un langage interprété sans compilation.
- JavaScript est plus simple que Java.
- JavaScript est généralement moins sécurisé que Java.

## 3.7 HTTP : HyperText Transfer Protocol

### 3.7.1 Fonctionnement

HTTP (HyperText Transfer Protocol) est le protocole central du Web. Il repose sur un modèle client-serveur où :

- Le client envoie des requêtes pour récupérer des objets Web.
- Le serveur répond avec les objets demandés.

#### HTTP utilise TCP

- le client établit une connexion TCP (crée un socket) avec le serveur, port 80



FIGURE 3.7 – Interaction Client/Serveur dans HTTP

- le serveur accepte la connexion TCP du client
- Les messages HTTP sont échangés entre le navigateur (client HTTP) et le serveur web.
- Fermeture de la connexion TCP

#### HTTP est « sans état » (Stateless)

- Le serveur ne conserve aucune information sur les demandes précédentes du client, chaque requête est donc traitée indépendamment des précédentes.
- C'est fait pour une question de gestion des ressources.

#### 3.7.2 Connexions HTTP : 2 types

Il existe principalement deux types de connexions HTTP : Persistant et Non-Persistant.

##### *HTTP Non-persistant*

1. Ouverture d'une connexion TCP
2. au plus un objet envoyé sur la connexion TCP
3. Fermeture de la connexion TCP

le téléchargement de plusieurs objets nécessite plusieurs connexions

##### *HTTP Persistant*

1. Ouverture d'une connexion TCP
2. plusieurs objets peuvent être envoyés via une seule connexion TCP entre le client et ce serveur
3. Fermeture de la connexion TCP

FIGURE 3.8 – Connexions HTTP

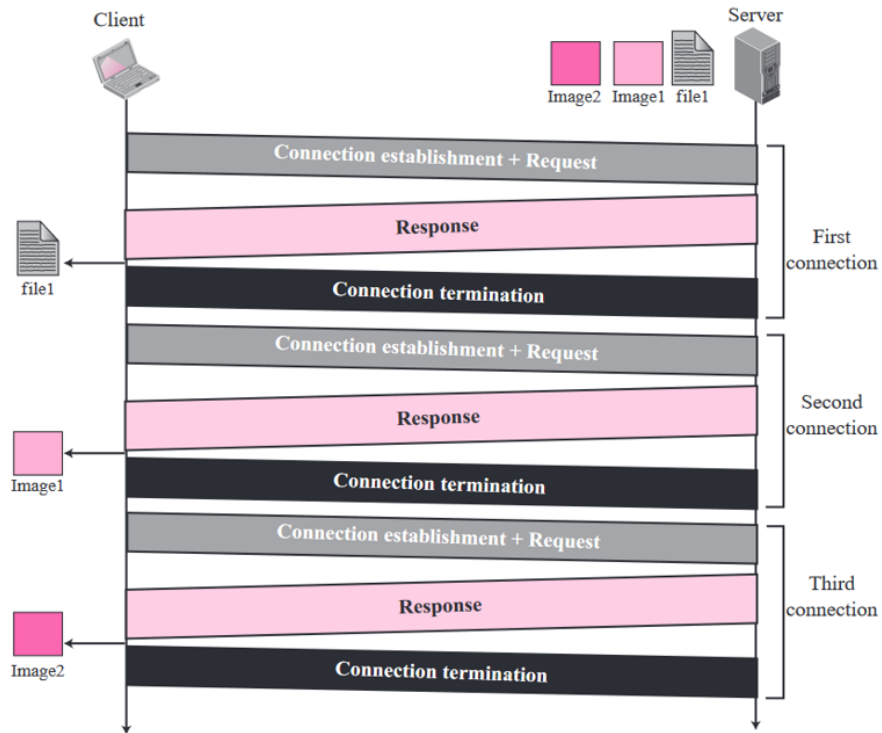


FIGURE 3.9 – HTTP non-persistent

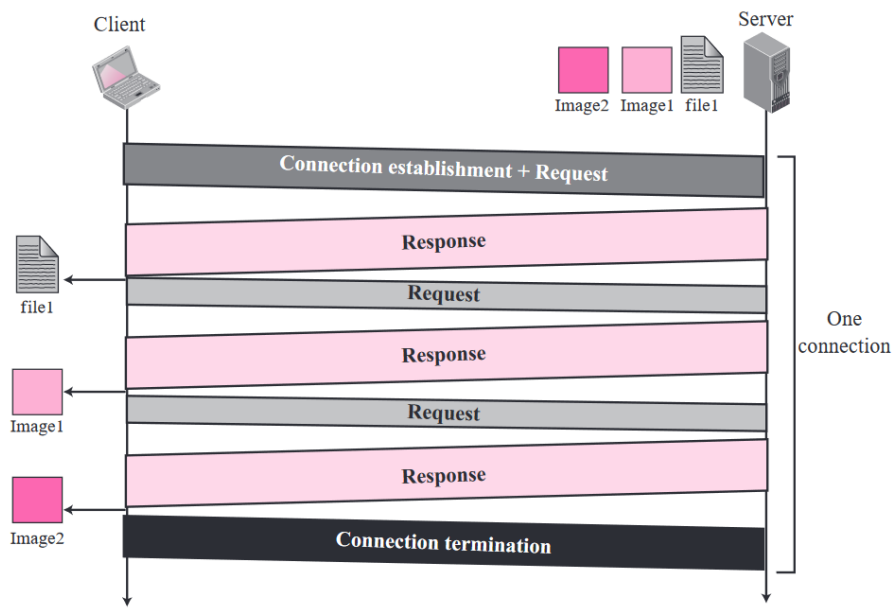


FIGURE 3.10 – HTTP persistent

### 3.7.3 Messages HTTP

Les échanges HTTP utilisent deux types de messages :

### Message de requête HTTP

Une requête se compose d'une ligne de requête, d'un en-tête, et parfois d'un corps.

- **Ligne de Requête** : C'est la première ligne avec trois champs : méthode, URL et Version. La ligne se termine par un retour chariot suivi d'un caractère de nouvelle ligne (saut de ligne).
- **Lignes d'En-tête** : Après la ligne de requête peuvent suivre des lignes d'en-tête. Chaque ligne d'en-tête envoie des informations supplémentaires du client au serveur. Par exemple, le client peut demander que le document soit envoyé dans un format spécial. Chaque ligne d'en-tête a un nom et une valeur d'en-tête.
- **Corps de la requête** : Parfois, les requêtes HTTP peuvent nécessiter d'envoyer des données du client vers le serveur, par exemple pour les méthodes POST et PUT. Ces données sont placées dans le champ corps (body).

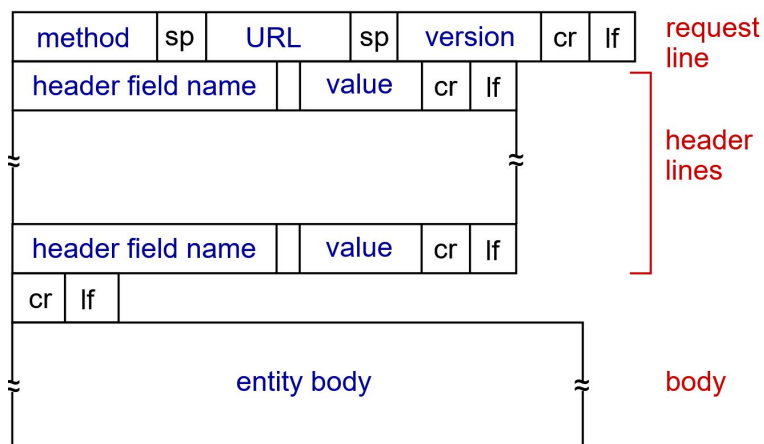


FIGURE 3.11 – Format d'une requête HTTP

Méthode	Action
GET	Demande un document au serveur
HEAD	Demande des informations sur un document, mais pas le document lui-même
POST	Envoie des informations du client au serveur

FIGURE 3.12 – Quelques méthodes HTTP

Header	Description
User-agent	Identifie le programme client
Accept	Indique le format multimédia accepté par le client
Accept-language	Indique la langue acceptée par le client
Authorization	Indique les permissions du client
Date	Indique la date actuelle

FIGURE 3.13 – Quelques Headers de requête HTTP

### Message de Réponse HTTP

Une réponse se compose d'une ligne d'état, de lignes d'en-tête, et parfois d'un corps.

— **Ligne d'État** : Trois champs séparés par des espaces.

- Version HTTP (par exemple 1.1).
- code d'état indiquant la réussite, la redirection, l'erreur client, ou l'erreur serveur (ex : 404).
- Phrase d'état qui explique le code en texte.(ex : "not found").

— **Lignes d'en-tête** : Permettent d'inclure des informations supplémentaires, comme la date, les cookies, etc.

— **Corps du message** : Contient le document à envoyer du serveur au client. Présent sauf en cas de message d'erreur.

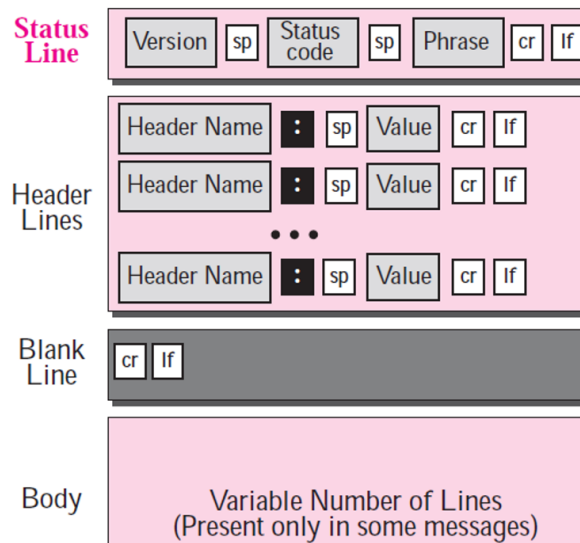


FIGURE 3.14 – Format d’une réponse HTTP

En-tête	Description
Date	Indique la date actuelle
Upgrade	Spécifie le protocole de communication préféré
Set-Cookie	Le serveur demande au client de sauvegarder un cookie
Content-Language	Spécifie la langue
Content-Length	Indique la longueur du document
Content-Type	Spécifie le type de média

FIGURE 3.15 – Quelques Headers de réponse HTTP

## 3.8 DNS : Domain Name System

Le DNS agit comme un annuaire téléphonique pour Internet, traduisant les noms de domaine (ex. `www.google.com`) en adresses IP numériques nécessaires aux machines pour communiquer.

### 3.8.1 DNS est une base de donnée hiérarchique

Il existe différents niveaux dans la hiérarchie DNS, chaque niveau de serveurs ayant un rôle bien spécifique.

- Serveurs racine (Root) : Ils constituent le sommet de la hiérarchie DNS et sont responsables de rediriger les requêtes vers les serveurs TLD. Ces serveurs ne stockent pas les noms de domaine complets, mais connaissent les emplacements des TLD.
- Serveurs TLD : Ils gèrent les extensions comme ".com", ".org", ou ".dz". Ces serveurs sont responsables de fournir les informations nécessaires pour localiser les serveurs d’autorité des domaines de second niveau.
- Serveurs d’autorité : Ils contiennent les enregistrements DNS spécifiques aux noms de domaine (comme "`www.exemple.com`"), indiquant les adresses IP associées ou

d'autres informations nécessaires à la résolution.

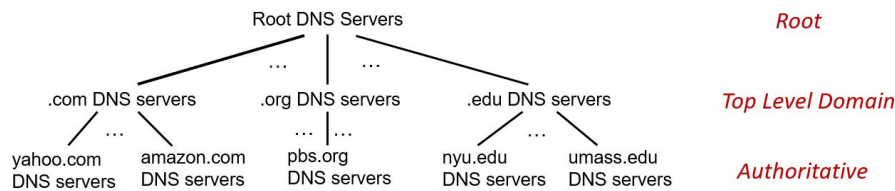


FIGURE 3.16 – Hiérarchie des serveurs DNS

### 3.8.2 Les étapes générales d'une résolution DNS

#### 1. Requête initiale

Lorsque vous saisissez une URL dans votre navigateur, celui-ci vérifie d'abord dans son cache local si une correspondance existe entre le nom de domaine et une adresse IP. Si le cache contient déjà cette information, le processus s'arrête ici.

#### 2. Consultation du serveur DNS récursif :

Si aucune information n'est disponible localement, le navigateur contacte le serveur DNS récursif de votre fournisseur d'accès Internet (FAI). Ce serveur agit comme un intermédiaire en recherchant l'adresse IP à votre place.

#### 3. Requête vers le serveur racine :

Si le serveur DNS récursif ne possède pas l'information dans son propre cache, il interroge l'un des 13 serveurs racine mondiaux. Ces serveurs racine ne contiennent pas directement l'adresse IP, mais savent orienter vers le bon serveur TLD (Top-Level Domain).

#### 4. Requête vers le serveur TLD :

Le serveur racine examine le domaine de premier niveau (TLD) de l'URL, comme ".com" ou ".org", et redirige la requête vers le serveur TLD correspondant. Ce serveur fournit des informations sur les serveurs d'autorité associés au domaine spécifique.

#### 5. Requête au serveur d'autorité :

Le serveur TLD transmet la requête au serveur d'autorité responsable du domaine, par exemple "google.com". Ce serveur contient les enregistrements DNS qui associent le nom de domaine à son adresse IP.

6. Transmission de l'information :

Le serveur d'autorité renvoie l'adresse IP correspondante au serveur DNS récursif, qui la stocke temporairement dans son cache pour des requêtes futures similaires.

7. Résolution finale :

Le serveur DNS récursif renvoie l'adresse IP au navigateur, qui peut alors stocker l'information dans le cache local et établir une connexion directe avec le serveur Web correspondant pour récupérer le contenu du site.

---