



Université de BATNA 2
Département d'informatique



1. INTRODUCTION

1.2 COMMENT FONCTIONNE LE WEB

Technologies Web, M1, RSD

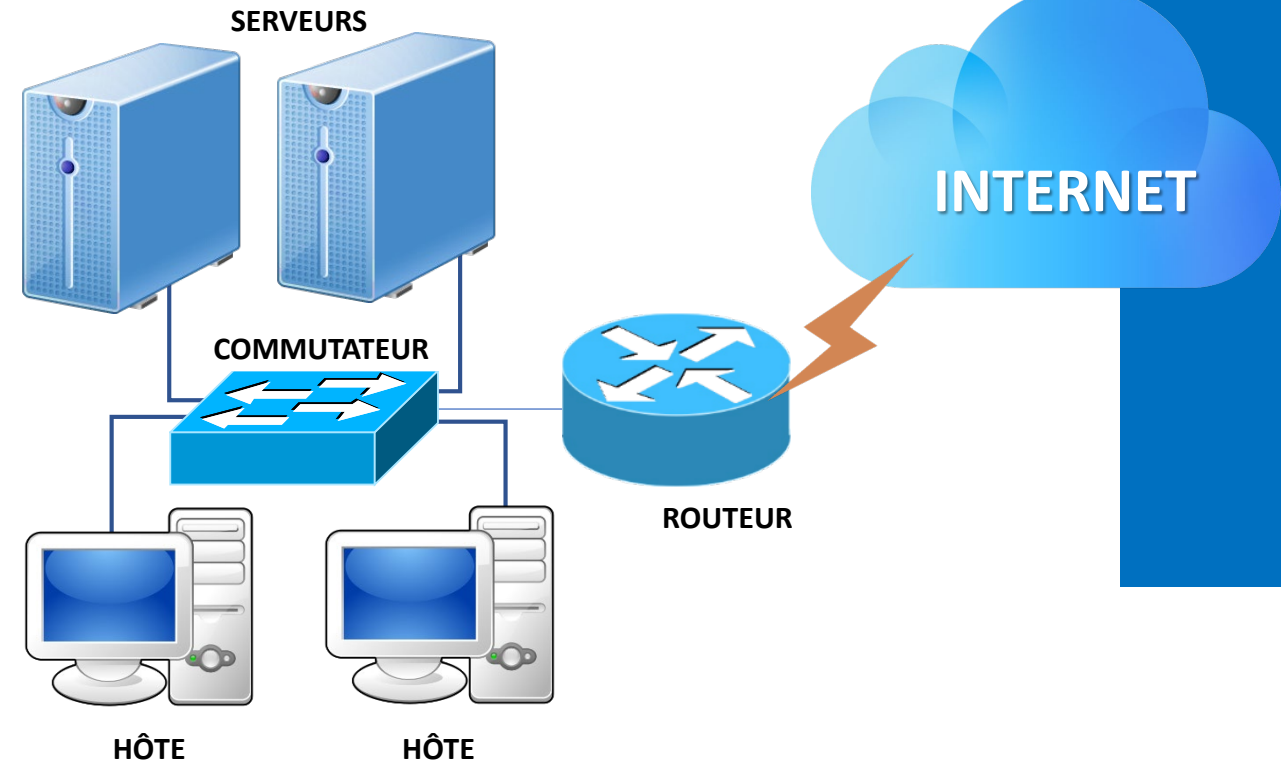
TABLE DE MATIÈRE

1. PRÉSENTATION DU RÉSEAU
2. LE MODÈLE CLIENT/SERVEUR
3. HTTP
4. WWW
5. INTÉRACTION CLIENT SERVEUR
6. HTTP REQUEST
7. HTTP RESPONSE
8. HTTPS

LE WEB PRÉSENTATION DU RÉSEAU

- Un réseau se compose d'un ensemble de périphériques finaux (**clients et serveurs**) interconnectés dans le but de **communiquer** et de **partager** des ressources.
- Les réseaux varient **en taille**

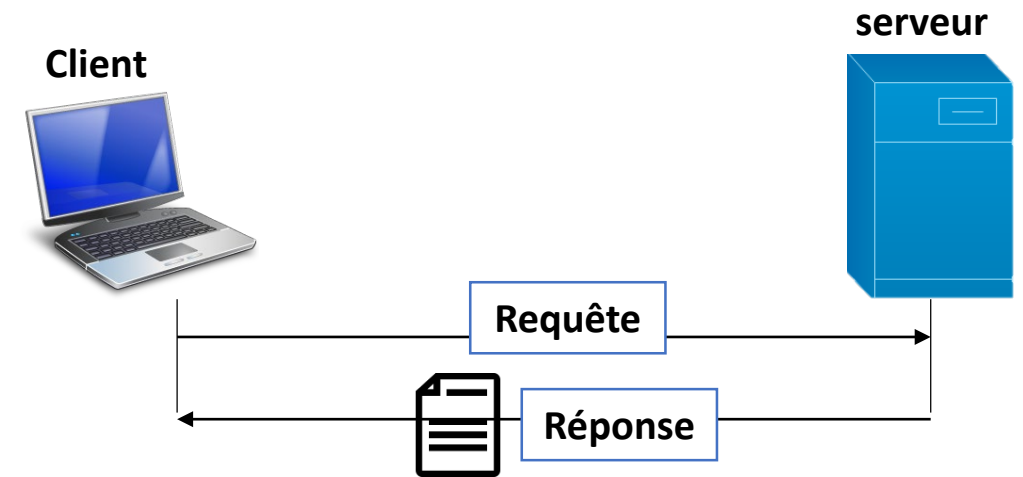
10 cm	0,01 Mètre carré	<i>Near Field Communication (NFC)</i>
1 m	Mètre carré	<i>Personal Area Network (PAN)</i>
10 m	Pièce dans un immeuble	<i>Local Area Network (LAN)</i>
100 m	Immeuble (building)	
1 Km	Campus	
10 Km	Ville	<i>Metropolitan Area Network (MAN)</i>
100 Km	Pays	<i>Wide Area Network (WAN)</i>
1000 Km	Continent	
10000 Km	Planète	<i>Internet</i>



- Composants communs d'un réseau:
 - Les serveurs
 - Les clients (ordinateurs)
 - Equipements réseau intermédiaires (routeur, commutateur, point d'accès, ...)

LE MODÈLE CLIENT/SERVEUR

- Les périphériques finaux peuvent jouer le rôle de client, ou de serveur, selon **les logiciels qu'ils exécutent**
- **Client**: hôte équipé de logiciel lui permettant de demander un certain type de service (tel qu'un accès à un fichier ou à une base de données) au serveur.
- **Serveur**: hôte exécutant un service en arrière plan, lui permettant de répondre à la demande du client et lui transmettre les résultats par l'intermédiaire d'un réseau.



Exemple

Client Web

- Exécute généralement **un navigateur web** (client) tel que Firefox
- Utilise HTTP
- **Demande** de pages web à un serveur
- **Reçoit** des pages web et des fichiers d'un serveur

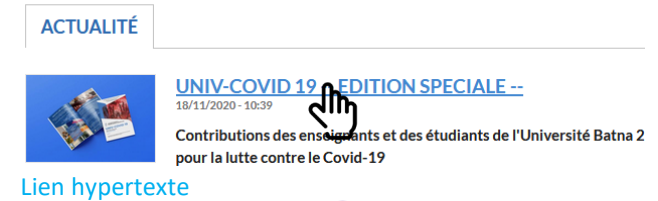
Serveur web

- Exécute un **service web** (Apache ou IIS)
- Utilise HTTP
- Reçoit une demande pour la page web
- Répond à la demande et transmet le code d'état, la page web, et fichiers associés au client

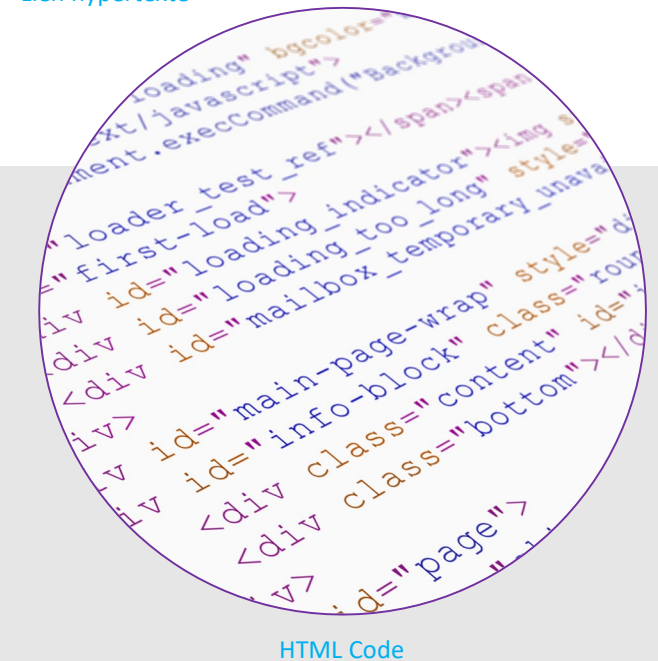
LE WEB

WWW

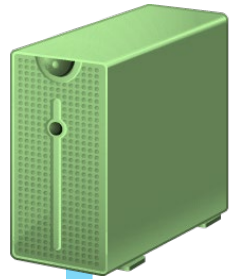
- Le Web est la collection de machines (serveurs Web) sur Internet qui fournissent des informations, notamment des documents **HTML**, via le protocole **HTTP**.
- Le Web prend en charge **les liens 'hypertexte'** (texte contenant des liens)
 - Communication via le protocole de transport hypertexte (**HTTP**)
 - Représentation des documents à l'aide du langage de balise hypertexte (**HTML**)



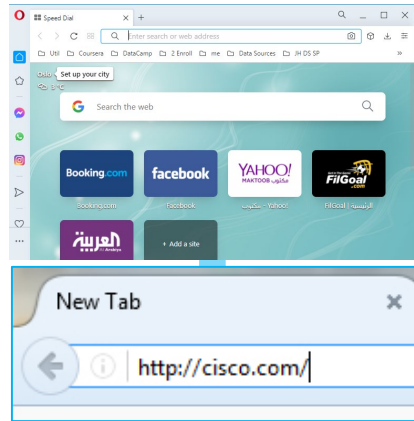
- Les machines qui accèdent aux informations sur le Web sont appelées **clients** Web. Un **navigateur** Web est un logiciel utilisé par un utilisateur final pour accéder au Web



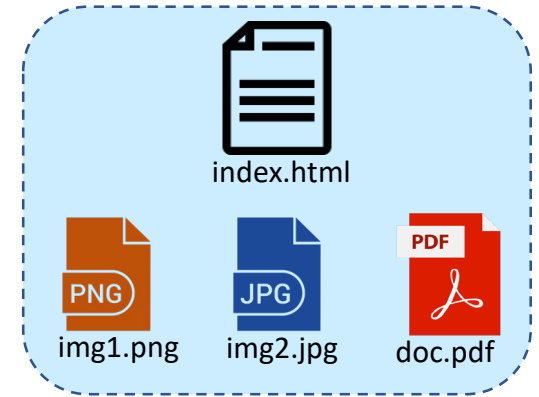
SERVEUR DNS



NAVIGATEUR



SERVEUR WEB



HTTP INTÉRACTION CLIENT SERVEUR

2. Requête DNS: résoudre l'URL

3. Réponse DNS: 23.43.193.127

1. saisir une URL

4. Le navigateur envoie une requête HTTP

5. Le serveur recherche le fichier en question

6. Le serveur renvoie une réponse HTTP contenant le fichier

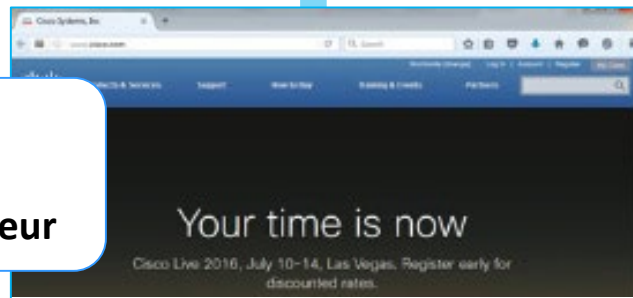
index.html

7. Le navigateur analyse le document et contacte à nouveau le serveur pour chaque objet joint (image, fichier PDF ...)

8. Le navigateur envoie des requêtes HTTP

9. Le serveur renvoie les objets demandés

10. la page est assemblée dans la fenêtre du navigateur



img1.png

img2.jpg

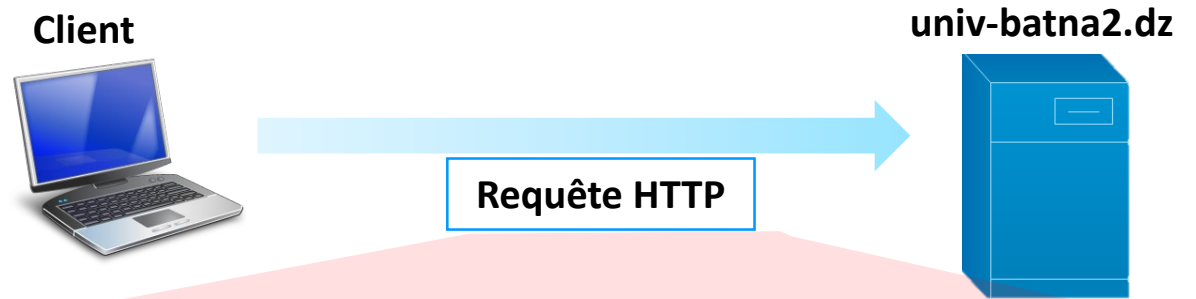
INTÉRACTION CLIENT SERVEUR

- Les informations transmises par HTTP sont souvent **entièrement textuelles**
- Nous pouvons utiliser le protocole **Telnet** d'Internet pour simuler la demande du navigateur et voir la réponse du serveur

```
Connect { $ telnet www.example.org 80
          { Trying 192.0.34.166...
            { Connected to www.example.com
              { (192.0.34.166).
                { Escape character is '^]'.
                  { GET / HTTP/1.1
                    { Host: www.example.org
                      {
                        { HTTP/1.1 200 OK
                          { Date: Thu, 09 Oct 2003 20:30:49 GMT
                            { ...
```

HTTP

HTTP REQUEST



Structure de la demande :

1. ligne de départ (*start line*)
2. champ(s) d'en-tête (*header field*) (un ou plusieurs)
3. ligne blanche
4. Corps du message (facultatif)

```
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
  [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
Host: univ-batna2.dz\r\n
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:82.0)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
\r\n
[Full request URI: http://univ-batna2.dz/]
[HTTP request 1/6]
...
```

ligne de départ

champ(s) d'en-tête

ligne blanche

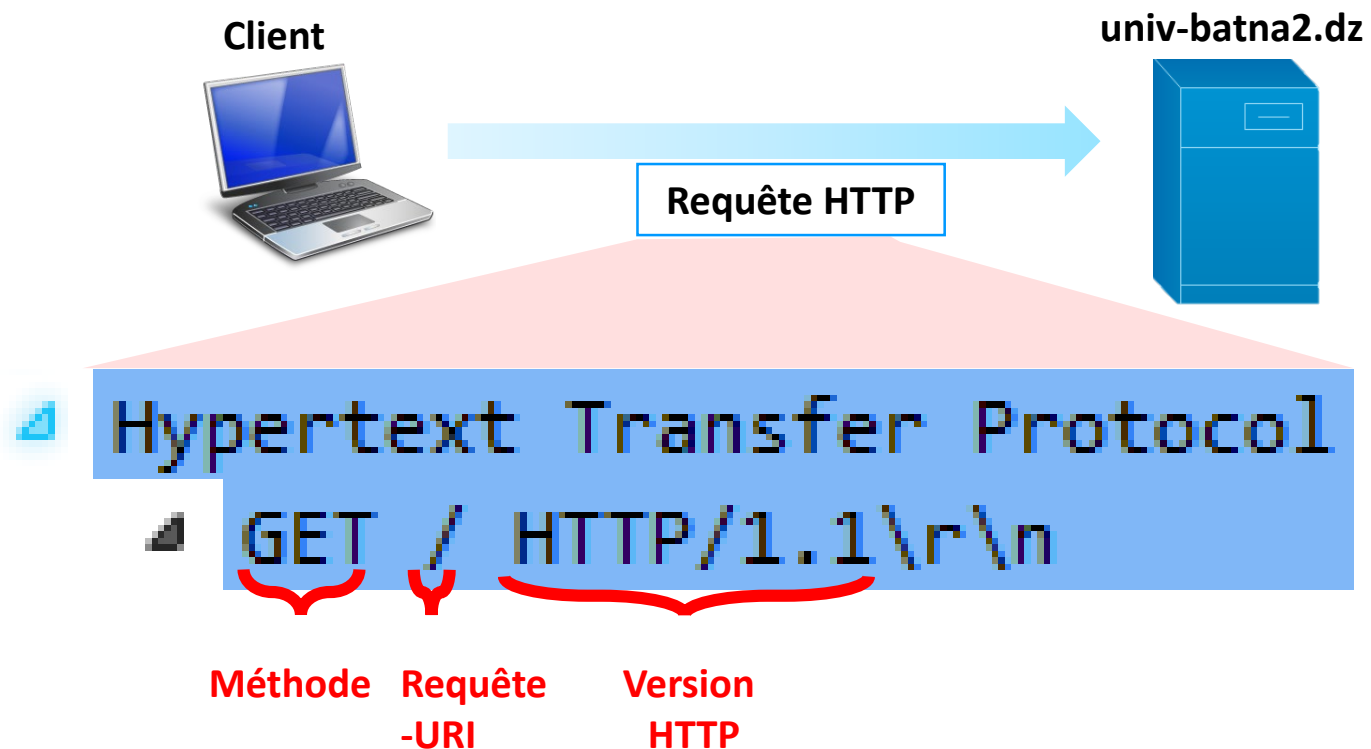
Corps du message

Ligne blanche

HTTP REQUEST: START LINE

Ligne de départ :Trois parties séparées par des **espaces** :

- Méthode de requête HTTP
- Requête-URI
- Version HTTP

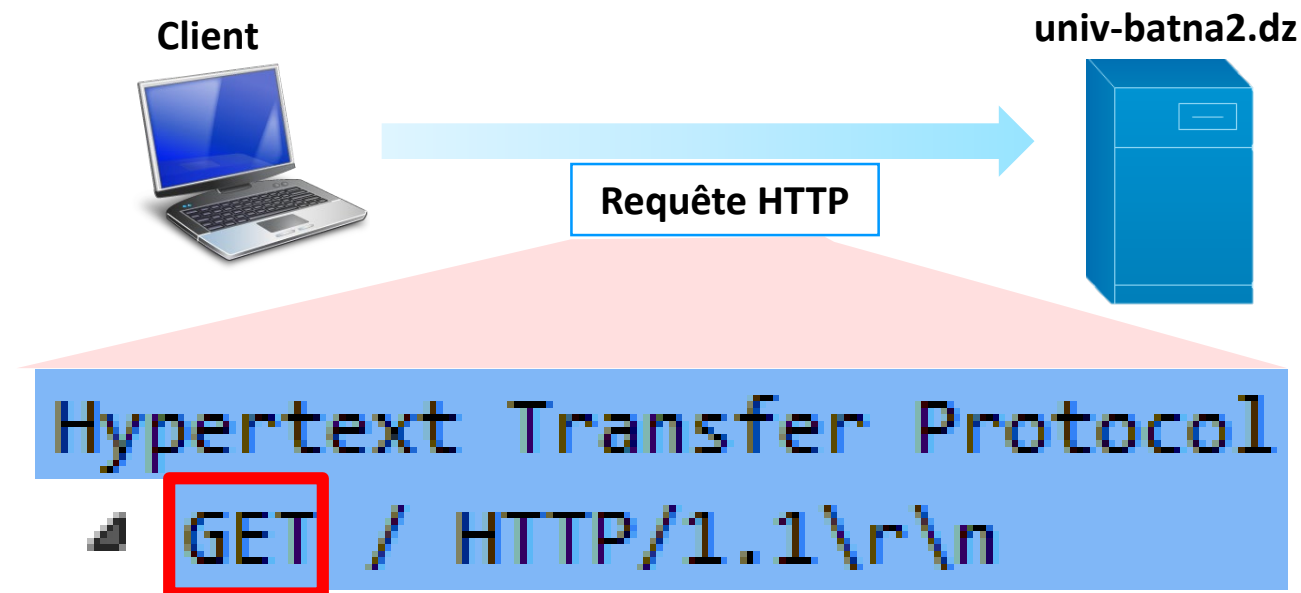


- ▶ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
Request Method: GET
Request URI: /
Request Version: HTTP/1.1

HTTP REQUEST: START LINE

Méthodes de demande communes :

- **GET**
 - Utilisé si un lien est cliqué ou l'adresse saisie dans le navigateur
 - Le corps de la requête GET est vides
- **POST**
 - Utilisé lorsque le bouton "Envoyer" est cliqué sur un formulaire
 - Informations saisies dans le formulaire sont contenues dans le corps de la demande
- **HEAD**
 - Seuls les champs d'en-tête sont à retourner dans la réponse (et non pas le corps)

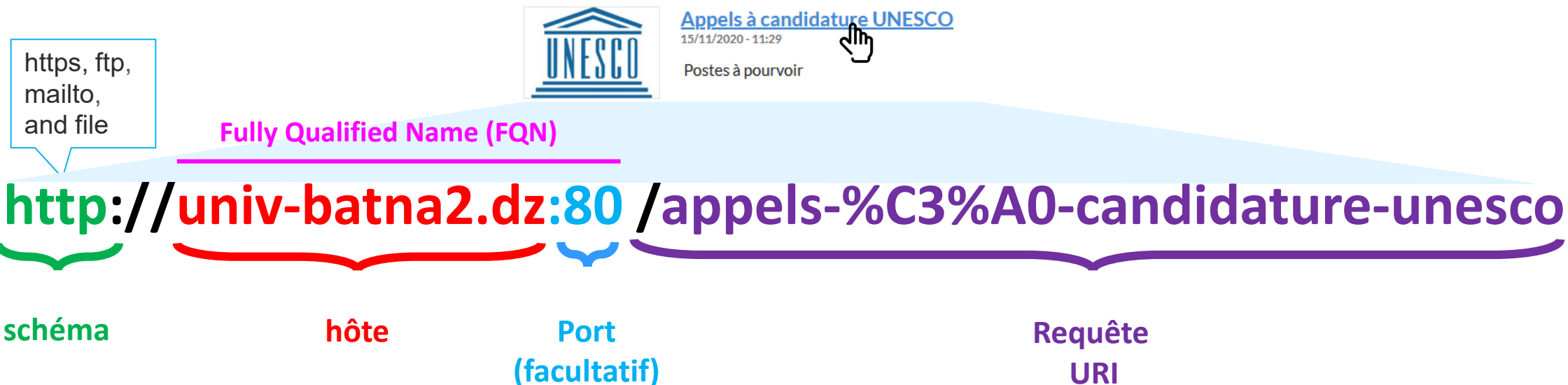


Autres méthode HTTP:

- **PUT** : demande que l'entité jointe soit stockée sous l'URI fourni (uploader un fichier)
- **DELETE**: supprime la ressource spécifiée.
- **OPTIONS**: renvoie les méthodes HTTP que le serveur prend en charge pour l'URL spécifiée.
- **PATCH** : applique des modifications partielles à une ressource.

HTTP REQUEST: START LINE

Uniform Resource Identifier (URI): une chaîne de caractères qui identifie sans ambiguïté une ressource particulière, soit par son emplacement, par un nom, ou par les deux.



L'URL Une URL est un type d'identifiant URI spécial qui indique également comment accéder à une ressource (Ex, son "emplacement" dans le réseau)

```

Hypertext Transfer Protocol
  GET /appels-%C3%A0-candidature-unesco HTTP/1.1\r\n
  [Expert Info (Chat/Sequence): GET /appels-%C3%A0-candidature-unesco HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /appels-%C3%A0-candidature-unesco
    Request Version: HTTP/1.1
    Host: univ-batna2.dz\r\n
  
```

HTTP REQUEST: EN-TÊTE

- **Structure du champ d'en-tête** : **nom du champ** : valeur du champ
- Les valeurs des champs peuvent contenir des **types MIME**, des **valeurs de qualité** et des **caractères génériques (*)**

1. Multipurpose Internet Mail

Extensions(MIME) : Convention pour la spécification du type de contenu d'un message

```
Accept : text/html, text/xml;q=0.9, text/plain ; q=0 . 8  
*/*;q=0.1
```

- Syntaxe: type de premier niveau/sous-type
Ex. text/html, image/jpeg

2. Valeurs de qualité : utilisé pour définir les préférences lors de la demande de plusieurs types de MIME.

```
Accept : text/html, text/xml;q=0.9, text/plain ; q=0 . 8  
*/*;q=0.1
```

Le client indique au serveur qu'il préférera avoir la réponse au format texte/html, en premier lieu. Si le serveur ne prend pas en charge ce format, il doit envoyer la demande au format xml sinon en plein texte. Si aucun des formats n'est disponible, la réponse sera envoyée dans le format supporté par le serveur

3. Caractère générique: utilisé pour spécifier la qualité 0,1 pour tout type de MIME non spécifié précédemment

```
Accept : text/html, text/xml;q=0.9, text/plain ; q=0 . 8  
*/*;q=0.1
```

HTTP REQUEST: EN-TÊTE

Champs d'en-tête communs :

- **host** : nom de l'hôte à partir de l'URL (obligatoire)
- **User-Agent** : type de navigateur qui envoie la demande
- **Accept** : Types de documents acceptables MIME
- **Connection** : la valeur "**close**" indique au serveur de fermer la connexion TCP après une demande/réponse unique. "keep-alive": plusieurs demandes/réponses en une connexion TCP.
- **Content-Type**: Type MIME du corps de la méthode POST, généralement application/x-www-form-urlencoded

```
Host: univ-batna2.dz\r\n
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64;
Accept: text/html,application/xhtml+xml,application/
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
\r\n
```

HTTP RESPONSE

Structure de la demande :

1. ligne d'état (*status line*)
2. champ(s) d'en-tête (*header field*) (un ou plusieurs)
3. ligne blanche
4. Corps du message (facultatif)

Ligne d'état

champ(s) d'en-tête

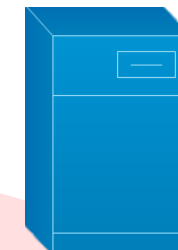
Ligne blanche

Corps du message

Client



univ-batna2.dz



Réponse HTTP

```

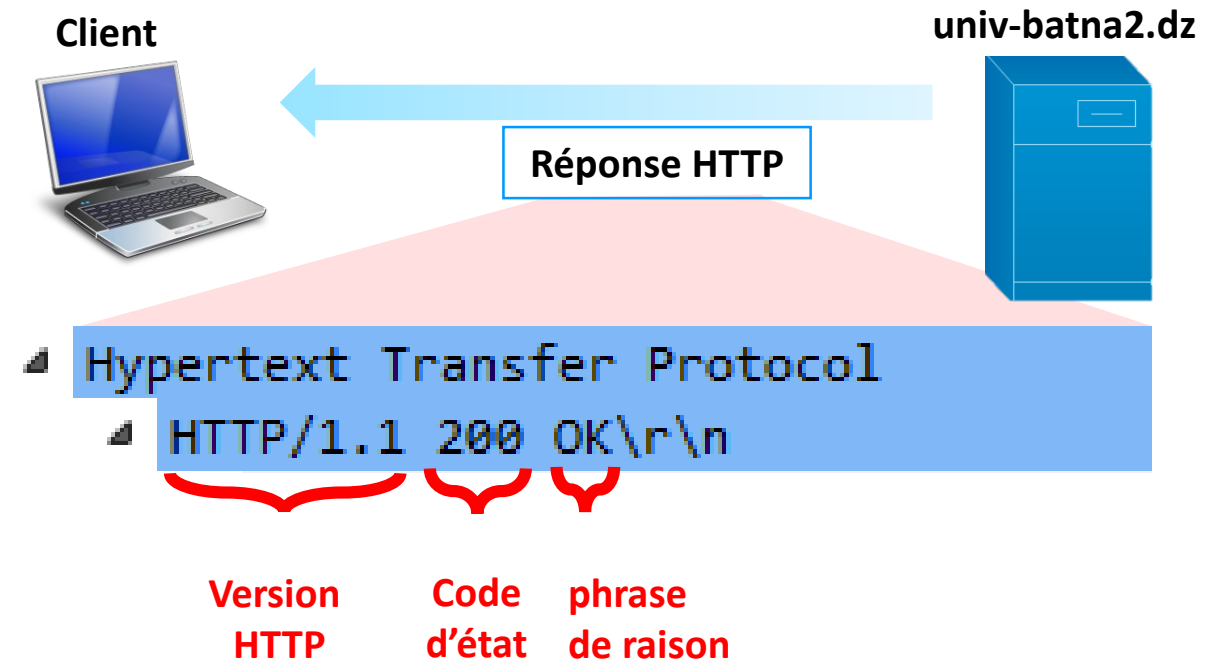
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Date: Tue, 24 Nov 2020 05:26:48 GMT\r\n
    Server: Apache\r\n
    Expires: Sun, 19 Nov 1978 05:00:00 GMT\r\n
    Cache-Control: no-cache, must-revalidate\r\n
    X-Content-Type-Options: nosniff\r\n
    Content-Language: en\r\n
    X-Frame-Options: SAMEORIGIN\r\n
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
    Content-Length: 13545\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=utf-8\r\n
    \r\n
  Line-based text data: text/html (1471 lines)
    <!DOCTYPE html>\n
    <html lang="en" dir="ltr"\n
  
```

Ligne blanche

HTTP RESPONSE: STATUS LINE

Ligne d'état: trois parties séparées dans l'espace :

- Version HTTP
- code d'état
- phrase de raison

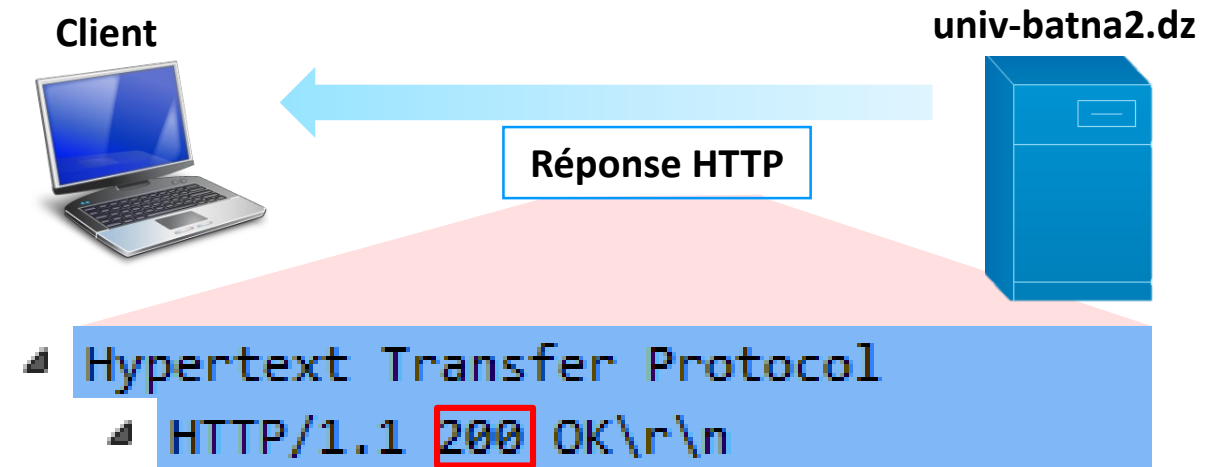


HTTP RESPONSE: STATUS LINE

Code d'état: Numéro à trois chiffres

Le premier chiffre est la classe du code de statut :

- 1 = Information
- 2 = Réussite
- 3 = Redirection (une autre URL est fournie)
- 4 = Erreur de client
- 5 = Erreur de serveur

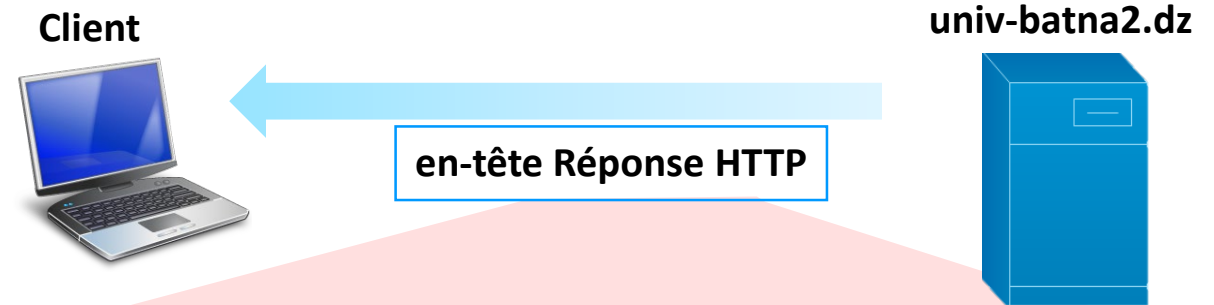


HTTP

HTTP RESPONSE: HEADER FIELD

Champs d'en-tête communs :

- **Connexion, Content-Type, Content-Length**
- **Date** : date et heure auxquelles la réponse a été générée (obligatoire)
- **Location** : URI alternatif si le statut est une redirection
- **Last-Modified**: date et heure de la dernière modification de la ressource demandée sur le serveur
- **Expires** : date et heure après lesquelles la copie de la ressource du client sera périmée
- **ETag** : un identifiant unique pour cette version de la ressource demandée (change si la ressource change)



```
Date: Tue, 24 Nov 2020 05:26:48 GMT\r\nServer: Apache\r\nExpires: Sun, 19 Nov 1978 05:00:00 GMT\r\nCache-Control: no-cache, must-revalidate\r\nX-Content-Type-Options: nosniff\r\nContent-Language: en\r\nX-Frame-Options: SAMEORIGIN\r\nVary: Accept-Encoding\r\nContent-Encoding: gzip\r\nContent-Length: 13545\r\nKeep-Alive: timeout=5, max=100\r\nConnection: Keep-Alive\r\nContent-Type: text/html; charset=utf-8\r\n\r\n
```

HTTP RESPONSE

- **Corps du message:** représente généralement le code HTML de la page demandée, retourné par le serveur. Ce code va être interprété puis affiché par le navigateur

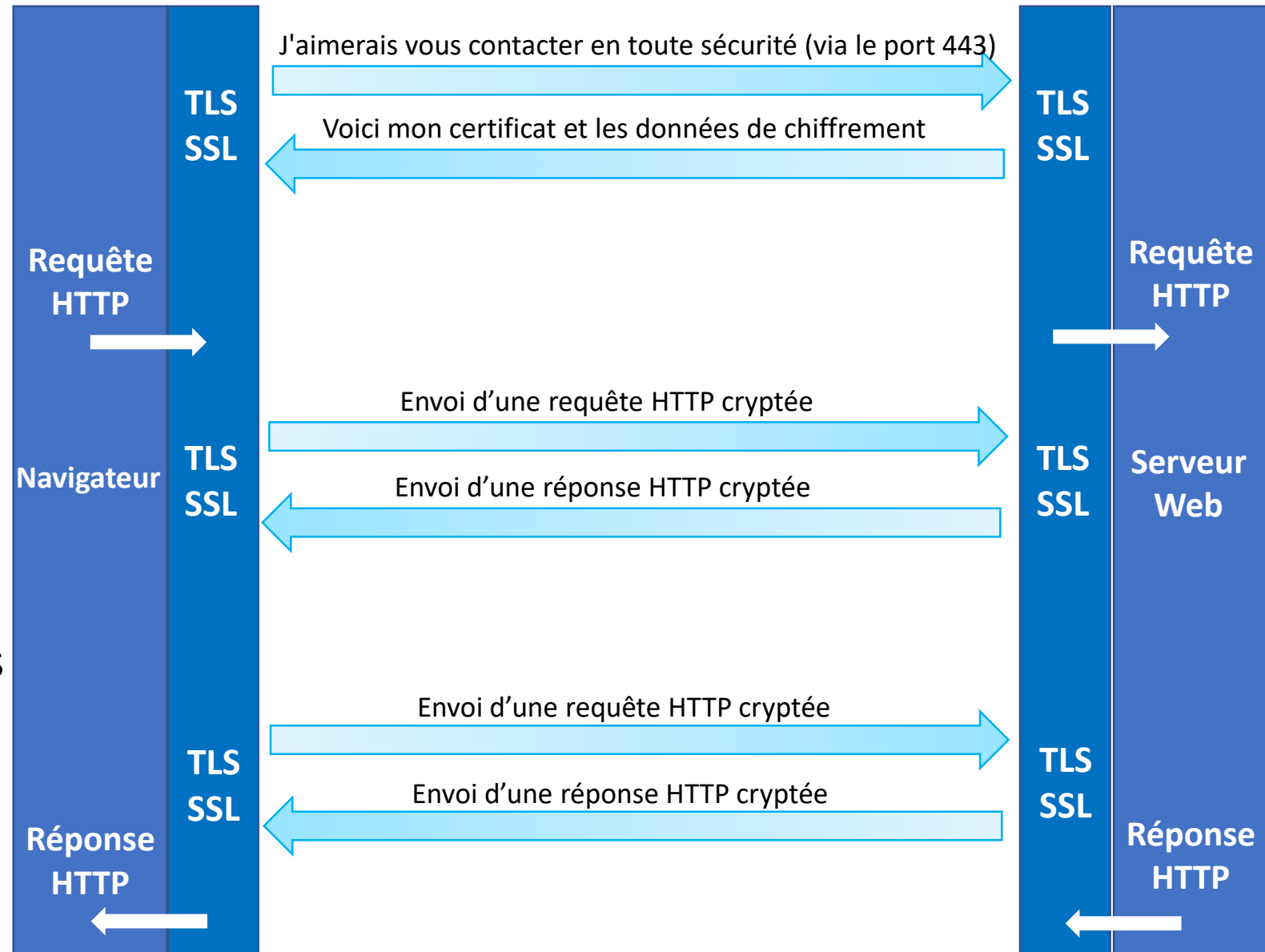
L'HTML sera le sujet du prochain chapitre de ce cours

```
Line-based text data: text/html (1270 lines)
<!DOCTYPE html>\n
<html lang="en" dir="ltr"\n
  xmlns:og="http://ogp.me/ns#\n
  xmlns:article="http://ogp.me/ns/article#\n
  xmlns:book="http://ogp.me/ns/book#\n
  xmlns:profile="http://ogp.me/ns/profile#\n
  xmlns:video="http://ogp.me/ns/video#\n
  xmlns:product="http://ogp.me/ns/product#\n
  xmlns:content="http://purl.org/rss/1.0/modules/content/"\n
  xmlns:dc="http://purl.org/dc/terms/"\n
  xmlns:foaf="http://xmlns.com/foaf/0.1/"\n
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#\n
  xmlns:sioc="http://rdfs.org/sioc/ns#\n
  xmlns:sioc_t="http://rdfs.org/sioc/types#\n
  xmlns:skos="http://www.w3.org/2004/02/skos/core#\n
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#">\n
\n
<head profile="http://www.w3.org/1999/xhtml/vocab">\n
<!-- Global site tag (gtag.js) - Google Analytics -->\n
<script async src="https://www.googletagmanager.com/gtag/js?id=UA-120527133-1">\n
<script>\n
  window.dataLayer = window.dataLayer || [];\n
  function gtag(){dataLayer.push(arguments);} \n
  gtag('js', new Date());\n
\n
  gtag('config', 'UA-120527133-1');\n
</script>\n
\n
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />\n
  <meta name="HandheldFriendly" content="true" />\n
  <meta name="MobileOptimized" content="width" />\n
  <meta name="viewport" content="width=device-width, initial-scale=1" />
```

HTTP

HTTPS

- Comme les messages HTTP circulent généralement sur un réseau public, les informations privées (telles que les numéros de carte de crédit, les mots de passes) doivent être **cryptées** pour éviter les écoutes
- Le schéma « **HTTPS** » d'URL indique au navigateur à utiliser le **chiffrement**.
- Normes de cryptage communes
 - Secure Socket Layer (**SSL**)
 - Transport Layer Security (**TLS**)



- “Uniform Resource Identifiers (URI): Generic Syntax”, Berners-Lee, Tim; Fielding, Roy T.; Masinter, Larry (January 2005).
- "Web Technologies - A Computer Science Perspective ", - J. Jackson , Pearson Education, Inc, 2007
- “Computer Networking, Sixth Edition”, James F. Kurose, Keith W. Ross, Pearson Education, Inc., 2013.

| **Merci** |