



Formateur



**Pr. O. EL MIDAOU**

Professor & Senior JAVA Software Engineer

PhD in Data Science - Web Geographic Information Retrieval GIR

# Plateforme

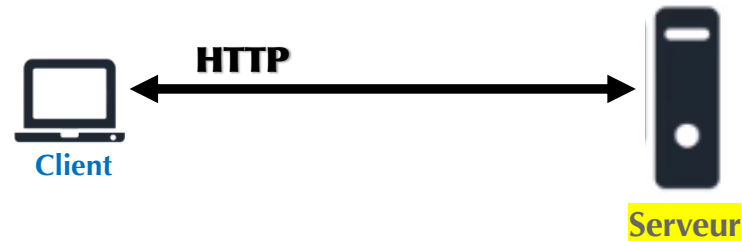
## Java EE



## Le protocole HTTP

## Architecture Client Serveur

### Protocole : HTTP



### HTTP :HyperText Tranfert Protocol

- Protocole qui permet au client de récupérer des documents du serveur
- Ces documents peuvent être statiques (contenu qui ne change pas : HTML, PDF, Image, etc..) ou dynamiques (Contenu généré dynamiquement au moment de la requête : PHP, JSP, ASP...)
- Ce protocole permet également de soumissionner les formulaires

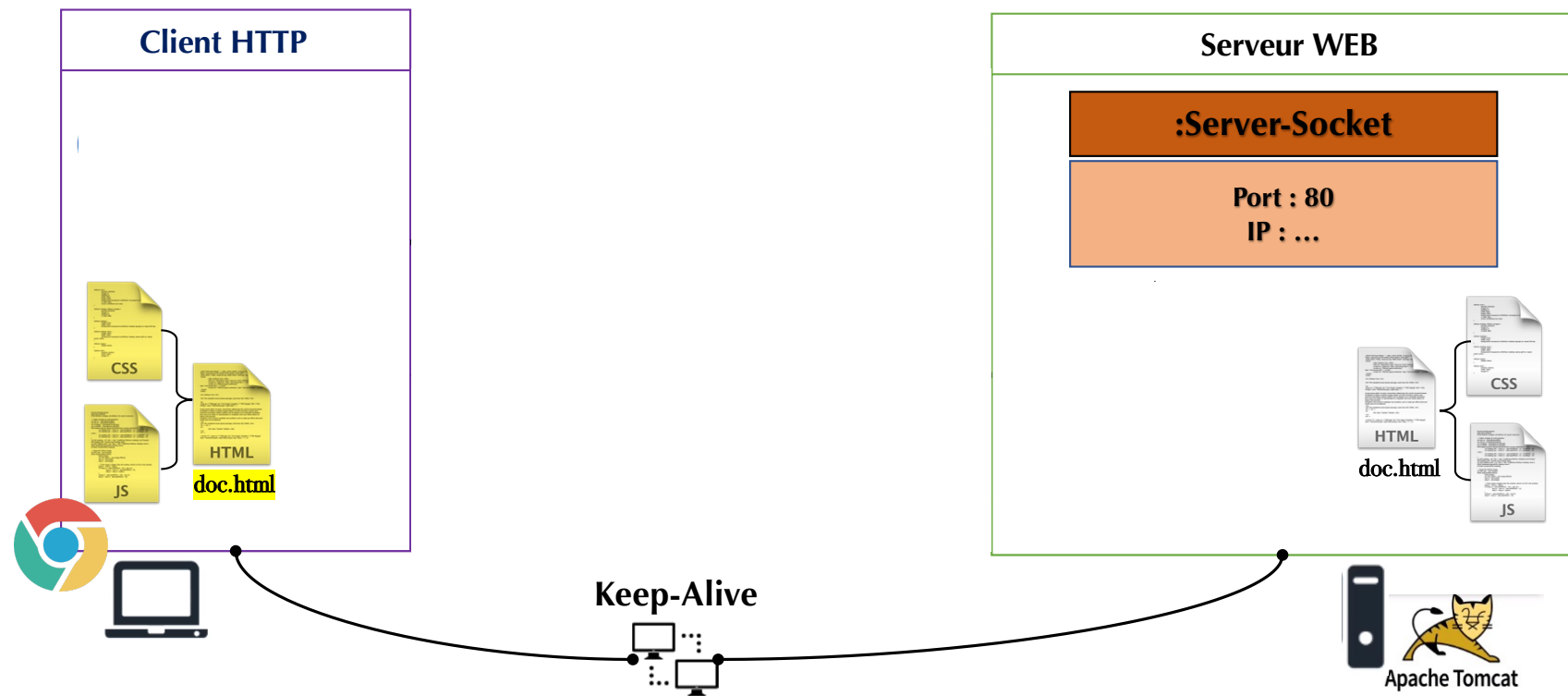
### Fonctionnement (très simple en HTTP/1.0)

- Le client se connecte au serveur (Créer une socket)
- Le client demande au serveur un document : Requête HTTP
- Le serveur renvoi au client le document (**status=200**)
- ou Le serveur renvoi une erreur (**status=404**) quand le document n'existe pas
- Déconnexion



## Architecture Client Serveur

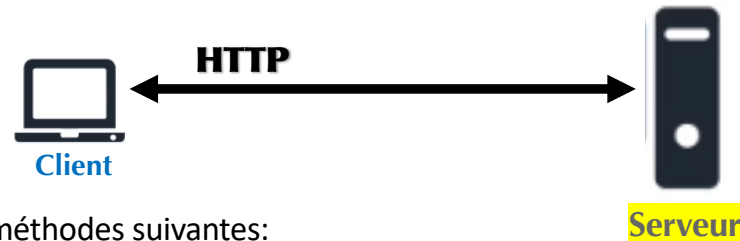
### Protocole : HTTP



## Architecture Client Serveur

### Protocole : HTTP

#### Méthodes HTTP



✓ Une requête HTTP peut être envoyée en utilisant les méthodes suivantes:

- **GET** : Pour récupérer le contenu d'un document
- **POST** : Pour soumissionner des formulaires (Envoyer, dans la requête, des données saisies par l'utilisateur )
- **PUT** pour envoyer un fichier du client vers le serveur
- **DELETE** permet de demander au serveur de supprimer un document.
- **HEAD** permet de récupérer les informations sur un document (Type, Capacité, Date de dernière modification etc...)

✓ Avec HTTP/1.1:

- **OPTIONS** : Permet au client HTTP d'interroger le serveur pour fournir les options de communications à utiliser pour une ressource ciblée ou un ensemble de ressources  
(Méthodes autorisées, Entêtes autorisées, Origines autorisés, ...)  
-> utilisé souvent comme requête de pré-vérification CROSS-ORIGINE.  
souvent pour des raisons de sécurité les navigateur refusent l'envoi de requête d'un domaine vers un autre à moins que si le domaine cible autorise cette opération



## Architecture Client Serveur

### Protocole : HTTP

#### Exemple d'une requête POST : HTTP

```
POST /login HTTP/1.1  
HOST : www.domaine.net  
ACCEPT : application/json  
Content-type : application/x-www-form-urlencoded  
Cookie : JSESSIONID : C4512152018CFDDOP345677
```

// Saut de ligne

```
username=admin & password=1234 & action=login
```

Entête de la requête

Corps de la requête



## Architecture Client Serveur

### Protocole : HTTP

#### Exemple d'une requête POST : HTTP

```
POST /login HTTP/1.0  
HOST : www.domaine.net  
ACCEPT : application/json  
Content-type : application/json  
Cookie : JSESSIONID : C4512152018CFDDOP345677
```

// Saut de ligne

```
{ "username" : " admin " , "password" : "1234" , "action" : " login" }
```

Entête de la requête

Corps de la requête



## Architecture Client Serveur

### Protocole : HTTP

#### Exemple d'une requête GET : HTTP

**GET** /login ?username=admin & password=1234 & action=ok HTTP/1.0

**HOST** : [www.domaine.net](http://www.domaine.net)

**ACCEPT** : application/json

**Content-type** : application/x-www-form-urlencoded

**Cookie** : JSESSIONID : C4512152018CFDDOP345677

// Saut de ligne

Entête de la requête

Corps de la requête



## Architecture Client Serveur

### Protocole : HTTP

#### Exemple d'une réponse : HTTP

**HTTP/1.1 200 OK**

**Date :** Wed, 05Feb02 15:02:01 GMT

**Server :** Apache/1.3.24

**Last-Modified :** Wed, 02Oct01 24:02:01 GMT

**Content-type :** application/json

**Content-length :** 77

**SET-Cookie :** JSESSIONID : C4512152018CFDDOP345677 expires : Sat, 10-Avr-2023 09:30

// Saut de ligne

```
[  
  { "id" : 1 , "TaskName" : "T1"}, { "id" : 2 , "TaskName" : "T2"},  
  { "id" : 3 , "TaskName" : "T3"}  
]
```

Entête de la réponse

Corps de la Réponse





## Architecture Client Serveur

### Protocole : HTTP

#### Codes de statu : HTTP

- ✓ Lorsque le serveur renvoie un document, il lui associe un code e statu renseignant ainsi le client sur le résultat de la requête (requête invalide, document non trouvé ...)
- ✓ Il existe cinq catégories de codes d'état de réponse HTTP : 1xx, 2xx, 3xx, 4xx et 5xx
  - **Information 1xx :**
    - 100 (Continue): Utilisé dans le cas où la requête possède un corps volumineux.
    - 101 (switching protocole) : Demander au client de changer de protocole.  
i.e. switching de HTTP 1.0 vers HTTP 1.1
  - **Succès 2xx :**
    - 200 (OK): Le document a été trouvé et son contenu suit...
    - 201 (Created) : Le document a été créé en réponse à un **PUT**
    - 202 (Accepted) : Requête acceptée, mais traitement non terminé.
    - 204 (No Response) : Le serveur n'a aucune information à renvoyer.
    - 206 (Partial Content) : une partie du document suit...



## Architecture Client Serveur

### Protocole : HTTP

#### Codes de statu : HTTP

- **Redirection 3xx :**
  - 301 (Moved): Le document a changé d'adresse d'une façon permanente
  - 302 (Found) : Le document a changé d'adresse temporairement.
  - 304 (Not modified) : Le document demandé n'a pas été modifié.
- **Erreurs du Client 4xx :**
  - 400 (Bad Request): La syntaxe de la requête est incorrecte...
  - 401 (Unauthorized) : Le client n'a pas les privilèges d'accès au document
  - 403 (Forbidden) : L'accès au document est interdit
  - 404 (Not Found) : Le document demandé n'a pu être trouvé
  - 405 (Method not allowed) : La méthode de la requête n'est pas autorisée.
- **Erreurs du Serveur 5xx :**
  - 500 (Internal Error): Une erreur inattendu est survenue au niveau du serveur...
  - 501 (Not implemented) : La méthode utilisé n'est pas implémentée.
  - 502 (Bad Gateway) : Erreur du serveur distant lors d'une requête proxy.



## Architecture Client Serveur

# Protocole : HTTP

## Entêtes : HTTP

- **Entêtes HTTP génériques :**
  - **Content-length** : Longueur en octets des données suivant les en-têtes
  - **Content-Type**: type des données qui suivent.
  - **Connection**: indique si la connexion TCP doit rester ouverte (Keep-Alive) ou être fermée (close).
- **Entêtes de la requête :**
  - **Accept.** : type que le client accepte et attend.
  - **Accept-encoding**: type d'encodage que j'accepte.
  - **Accept-language**: le code de la langue utilisé par le navigateur
  - **Cookie.** : Données de cookie mémorisées par le client.
  - **Host** : Hôte virtuel demandé.
  - **If-modified-since**: ne retourne le document que s'il a changé depuis la date indiquée
  - **If-none-match.** : ne retourne le document que s'il a changé.
  - **Referer** : URL de la page à partir de laquelle le document est demandé.
  - **User-agent**: nom et version du logiciel utilisé



## Architecture Client Serveur

### Protocole : HTTP

#### Entêtes : HTTP

- **Entêtes de la réponse :**
  - **Allowed.** : Méthodes HTTP autorisées pour cette URI (comme PORT).
  - **Date** : date et heure courante (d'envoi de la réponse) .
  - **Expires**: date d'expiration du document (pour le supprimer du cache)
  - **Last-modified.** : Date du dernière modification.
  - **Location** : Adresse du document lors d'une redirection.
  - **Etag**: numéro de version du document
  - **Pragma.** : données annexes pour le navigateur (i.e. no.cache).
  - **Server** : nom et version du serveur .
  - **Set-cookie**: permet au serveur d'écrire un cookie sur le disque du client.

