Université Constantine 2
جامعة قسنطينة 2

Développement Avancé d'Applications Web

– Chapitre 2 –

Le Composant Contrôleur (Servlet)

Dr Bouanaka Chafia
NTIC
chafia.bouanaka@univ-constantine2.dz

Etudiants concernés

Faculté/Institut	Département	Niveau	Spécialité
Nouvelles technologies	TLSI	Licence 3	Génie Logiciel (GL)

Servlets

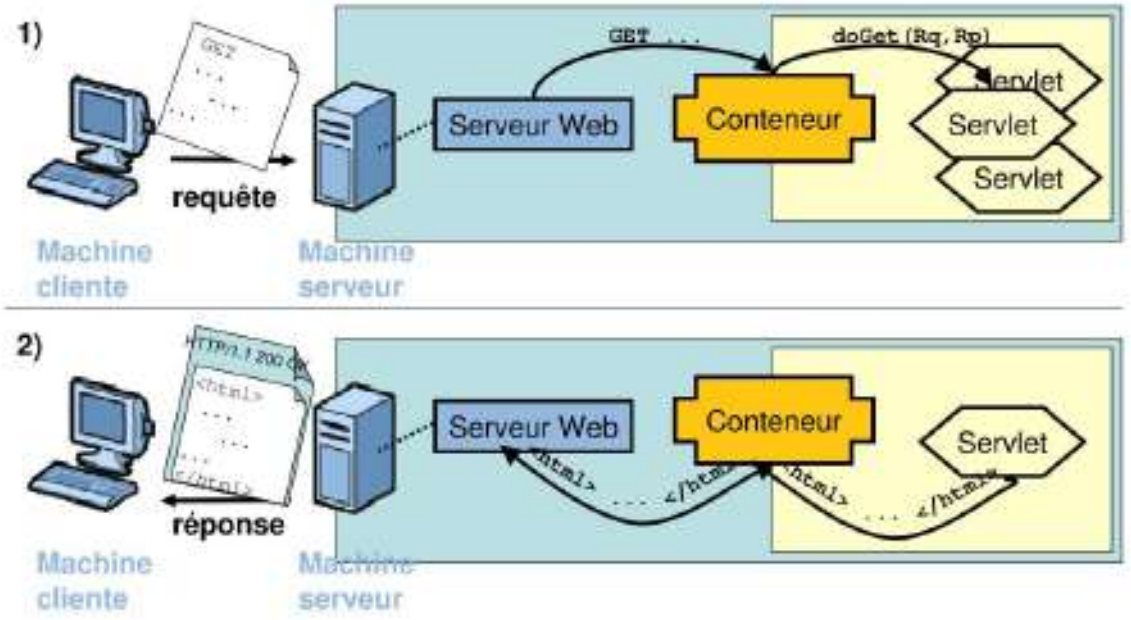
Définition d'une Servlet

- Une servlet est une classe Java, qui a la particularité de **permettre le traitement de requêtes et la personnalisation de réponses**
- C'est un programme "autonome" stocké dans un fichier .class sur le serveur web
- Exécutable avec tous les serveurs Web (Apache, IIS,...) ayant un moteur de servlet/JSP (e.g. Tomcat)
- Pour traiter de requêtes HTTP et générer des réponses dynamiques comparables aux CGI(Common Gateway Interface)
- Permet d'étendre les fonctionnalités de base d'un serveur HTTP

Servlets

Rôles d'une Servlet

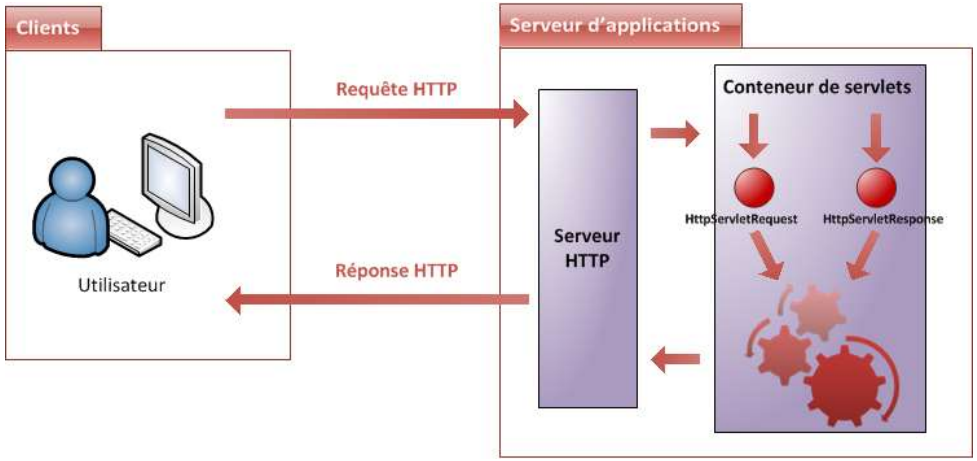
- Recevoir les données envoyées par l'utilisateur via un formulaire, du conteneur Web
- Déclencher les traitement métier demandés et recevoir les résultats
- Identifier les pages HTML dynamiques pour afficher les résultats



Rôles d'une Servlet

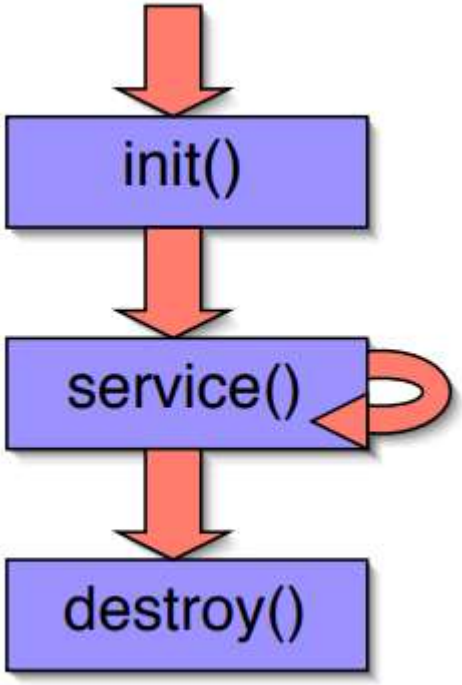
Recevoir les données d'entrées et préparer la réponse

- Lorsque le conteneur web reçoit une URL appelant une servlet, il crée deux nouveaux objets :
  - HttpServletRequest : cet objet contient la requête HTTP, et donne accès à toutes ses informations, telles que les en-têtes (headers) et le corps de la requête.
  - HttpServletResponse : cet objet initialise la réponse HTTP qui sera renvoyée au client, et permet de la personnaliser, en initialisant par exemple les en-têtes et le corps (nous verrons comment par la suite).
- Le conteneur Web, crée ensuite une instance de la servlet demandée et lui transmet les deux objets (Request/Response)



Servlets

Cycle de vie d'une servlet



- première requête de ce nom : chargement de la classe dans la JVM, instanciation, exécution de la méthode `init()`
- toute requête :
  - création des objets requêtes et réponses
  - création d'un thread. Exécution de la méthode `service()` ou d'une sous-méthode.
- fin de vie : appel à `destroy()` et destruction de l'objet.

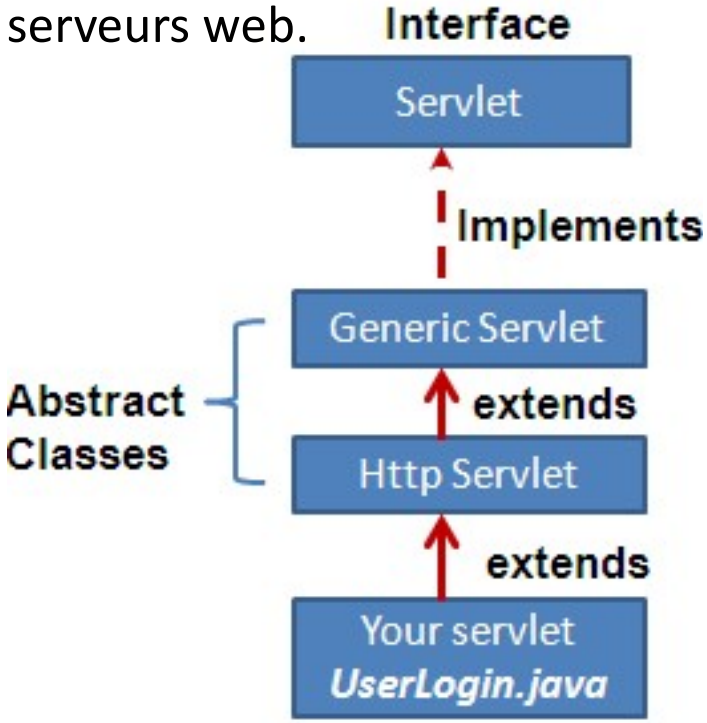
Servlets

Cycle de vie d'une servlet

- Les méthodes **init()**, **service()** et **destroy()** assurent le cycle de vie de la servlet en étant respectivement appelées lors de la création de la servlet, lors de son appel pour le traitement d'une requête et lors de sa destruction.
- La méthode **init()** est appelée par le conteneur web juste après l'instanciation de la servlet.
- La méthode **service()** ne peut pas être invoquée tant que la méthode **init()** n'est pas terminée.
- La méthode service réalise le traitement demandé
- La méthode **destroy()** est appelée juste avant que le conteneur détruise la servlet : cela permet de libérer les ressources allouées dans la méthode **init()** tel qu'un fichier ou une connexion à une base de données.

Structure de base d'une servlet

- L'API **servlet** est une extension du jdk de base, elle est regroupée dans des packages préfixés par **javax**
- L'API **servlet** regroupe un ensemble de classes dans deux packages :
  - **javax.servlet** : contient les classes pour développer des serlvets génériques indépendantes d'un protocole
  - **javax.servlet.http** : contient les classes pour développer des servlets qui reposent sur le protocole http utilisé par les serveurs web.



Structure de base d'une servlet

- La servlet définie par un concepteur est une sous-classe de **HttpServlet** dans laquelle on surcharge les méthodes **doGet()** ou **doPost()**, en fonction du mode d'envoi des données utilisé(par GET ou POST).
- Ces deux méthodes prennent deux arguments : HttpServletRequest et **HttpServletResponse**.
- **HttpServletRequest** renferme des méthodes grâces auxquelles on détermine des informations sur les données entrantes (les données d'un formulaire, les en-têtes d'une requêtes http ou le nom de l'ordinateur du client).
- **HttpServletResponse** permet de spécifier les informations renvoyées, comme le code d'état http, les en-têtes de la réponse (content-type, set-cookies,…).

Structure de base d'une servlet

Entete d'une servlet

protected void service (HttpServletRequest req ,
HttpServletResponse resp) throws ServletException , java .io. IOException

Structure de base d'une servlet

strucuture d'une servlet

```
public class SimpleServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException
    {
        // Lire les paramètres de la requête du client à partir de l'objet Request
        // traiter la requête
        // Répondre en construisant l'objet Response
    }
}
```

Structure de base d'une servlet

- L'interface **ServletRequest** définit plusieurs méthodes qui permettent d'obtenir des données sur la requête du client.
- L'objet **HttpServletRequest** (hérite de ServletRequest) encapsule la requête HTTP et fournit des méthodes pour accéder à toutes les informations nécessaires sur l'environnement du serveur.
- Parmi les méthodes proposées par ServletRequest :
- **setAttribute()** : permet de déposer un paramètre dans l'objet Request
- **getAttribute()** : permet de récupérer un paramètre de l'objet Request

Structure de base d'une servlet

- L'interface **ServletResponse** définit plusieurs méthodes qui permettent de fournir la réponse faite par la servlet suite à ces traitements.
- L'objet **HttpServletResponse** (hérite de ServletResponse) est utilisé pour construire un message de réponse HTTP renvoyé au navigateur client.
- Il contient les méthodes nécessaires pour définir : le type de contenu, en-tête et code de retour.
- Il contient aussi un flot de sortie pour envoyer des données (HTML ou autre) au navigateur.

Servlets

Structure de base d'une servlet

Exemple de servlet

```
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 {
        PrintWriter out = response.getWriter();
        out.println("Hello World");
}
}
```

Structure de base d'une servlet

Exemple de servlet

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloGet extends HttpServlet {
public void doGet(HttpServletRequest requete, HttpServletResponse reponse) throws
ServletException, IOException {
        String prenom = requete.getParameter("prenom");
        reponse.setContentType("text/html");
        PrintWriter out= reponse.getWriter();
        out.println("<html><body>");
        out.println("<h1>Bonjour " + prenom + " ! </h1>");
        out.println("</body></html>");
}}
```

Fonctionnalités d'une Servlet

- Le protocole HTTP est un protocole non connecté (on parle aussi de protocole sans états, en anglais stateless protocol), cela signifie que chaque requête est traitée indépendamment des autres et qu'aucun historique des différentes requêtes n'est conservé.
- Le serveur ne conserve pas les données d'un client, une fois la réponse envoyée.
- Ainsi le serveur web ne peut pas se "souvenir" de la requête précédente, ce qui peut poser problème dans certaines utilisations telles que le e-commerce, pour lequel le serveur doit mémoriser les achats de l'utilisateur sur les différentes pages.
- Le traitement des requêtes clientes est réalisé requête par requête

  le web n'offre par défaut aucun suivi de session.

Concept de session

Permettre au serveur de mémoriser les informations relatives au client.

Fonctionnalités d'une Servlet : l'objet Session

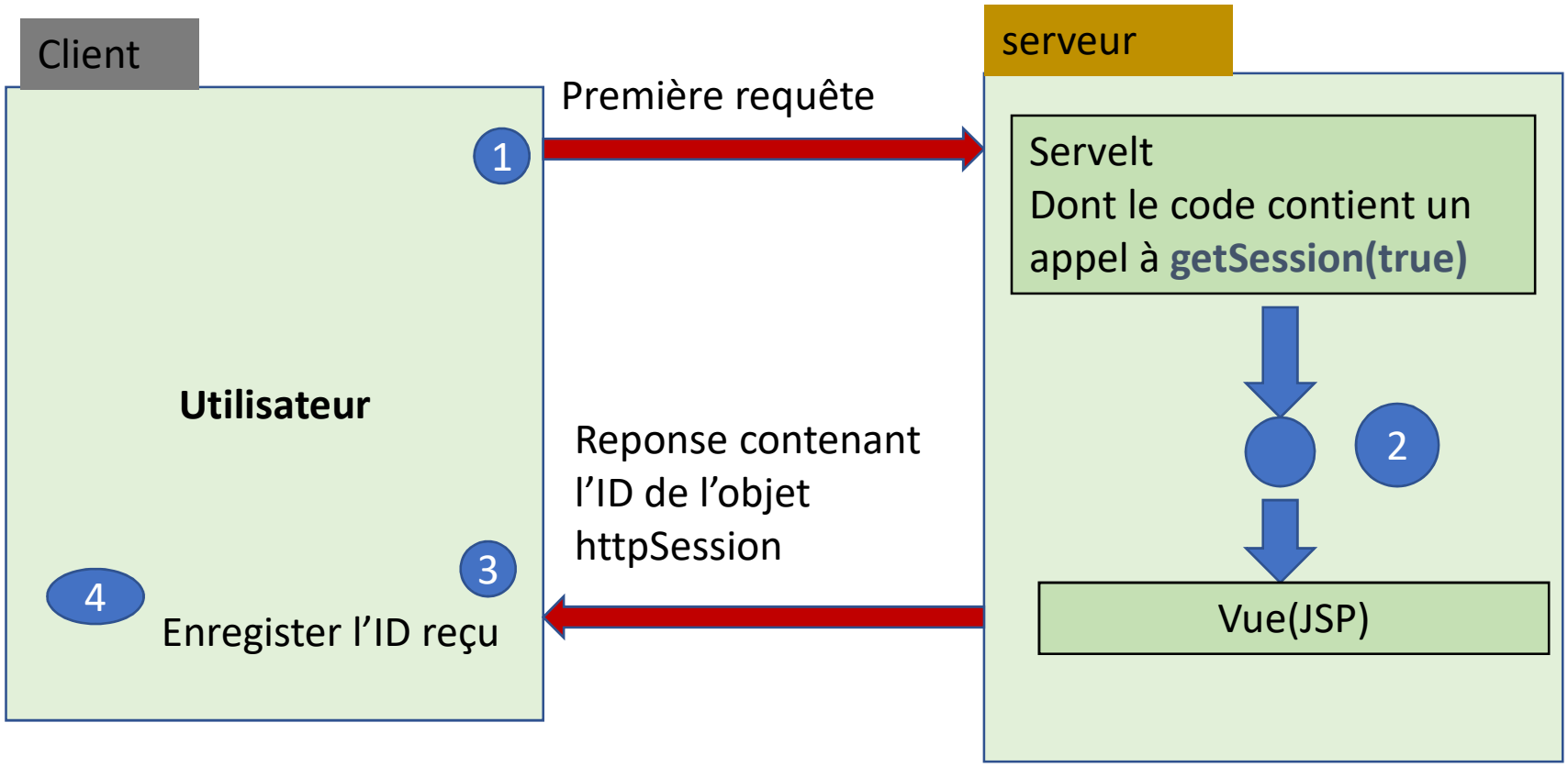
- Représente un espace mémoire alloué pour chaque utilisateur, pour sauvegarder des informations .
- Le contenu d'une session est conservé jusqu'à ce que l'utilisateur ferme son navigateur, reste inactif trop longtemps, ou il se déconnecte.
- Cette fonctionnalité est supportée par les servlets à l'aide de l'objet HttpSession obtenu à l'aide de request.getSession()

```
HttpSession session = request.getSession();
```

- L'identifiant de l'objet session est échangé à chaque fois entre le navigateur du client et le serveur web(voir schéma suivant)

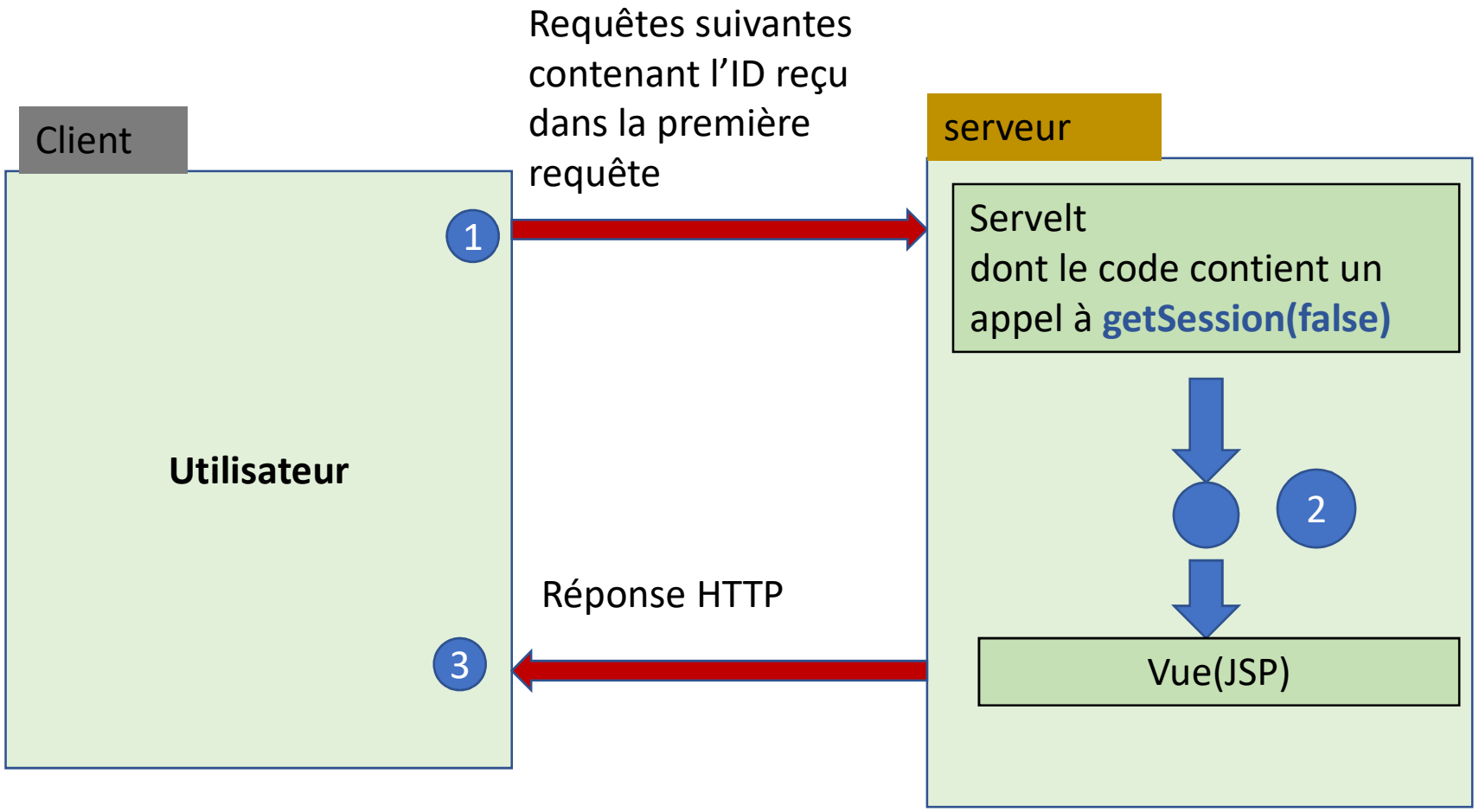
Fonctionnalités d'une Servlet : l'objet Session

- **Création et échange d'un objet Session**



Fonctionnalités d'une Servlet : l'objet Session

- **Création et échange d'un objet Session**



Fonctionnalités d'une Servlet : l'objet Session

L'objet session propose les méthodes suivantes :

void setAttribute(String name, Object value)

ajoute un couple (name, value) pour cette session

Object getAttribute(String name)

Retourne :

- l'objet associé à la clé name s'il le trouve
- ou null s'il ne le trouve pas

Session

L'objet **session** propose les méthodes suivantes :

void **setAttribute**(String name, Object value)

ajoute un couple (name, value) pour cette session

Object **getAttribute**(String name)

retourne

- l'objet associé à la clé name s'il le trouve
- ou null s'il ne le trouve pas