

Applications et Services WEB

JavaServer Pages

Erick Stattner

erick.stattner@univ-ag.fr

Maître de Conférences en Informatique

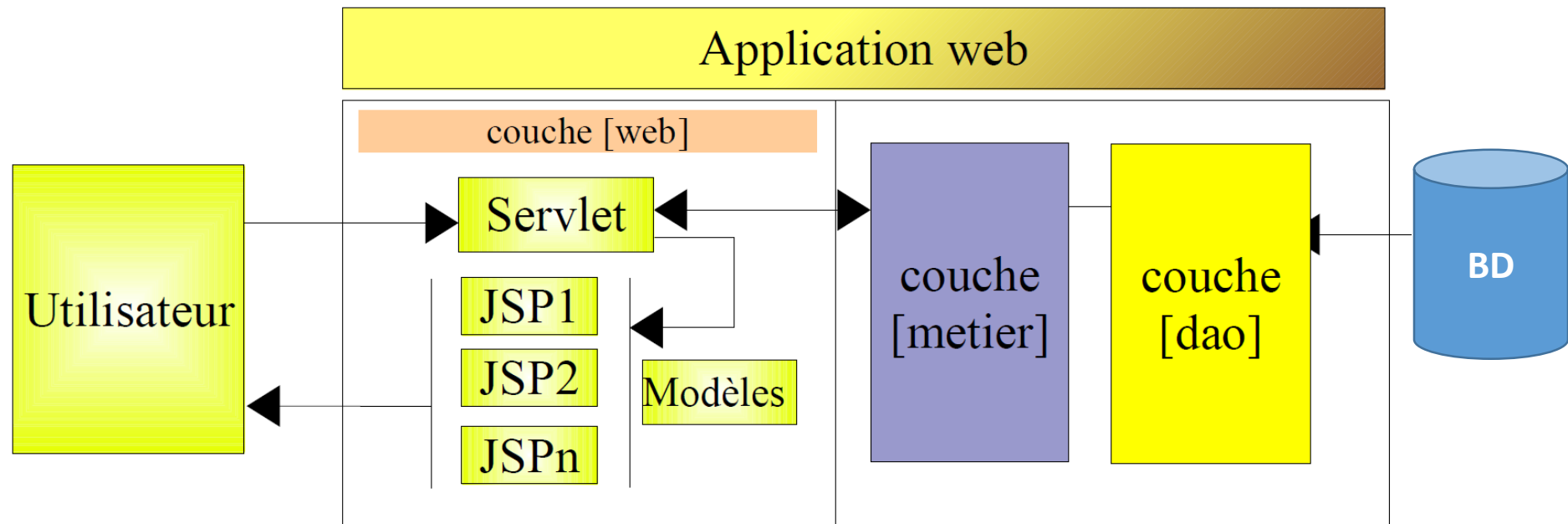
Université des Antilles



I. Introduction

Contexte

- La servlet ne génère pas directement la vue
- Séparer le fond et la forme
- Respecter le modèle MVC



Sommaire

1. **La technologie JSP**
2. Balises
3. Directives
4. Transmission de données

I. La technologie JSP

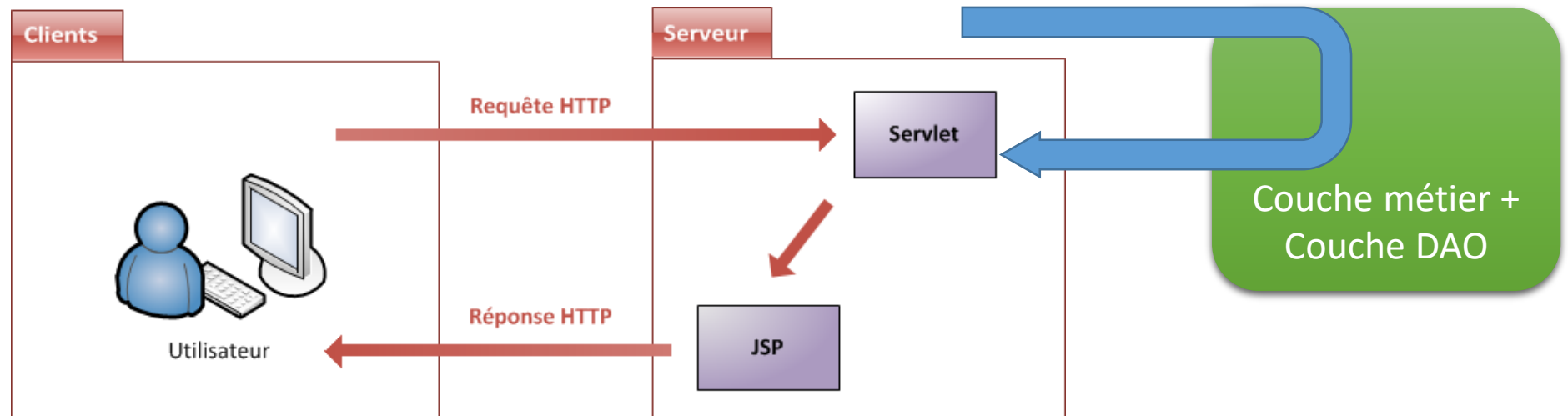
JavaServer Pages (JSP)

- Technologie JAVA qui permet de générer des pages WEB dynamiques
- Utilisé pour la vue des applications WEB
 - Très proche d'une page HTML
 - Exécutée sur le serveur
 - Page HTML finale générée et envoyée au client
 - Le client ne voit pas le code de la JSP
- Repose sur l'utilisation de balises XML
 - Appelées **actions/directives JSP**
- Permet d'utiliser du code JAVA pour générer la vue

I. La technologie JSP

Principe

- Lors d'une requête d'une client
 1. La servlet reçoit la requête
 2. La traite
(va éventuellement solliciter les données nécessaire au traitement)
 3. Fournit un affichage à travers la vue (JSP)



I. La technologie JSP

Principe

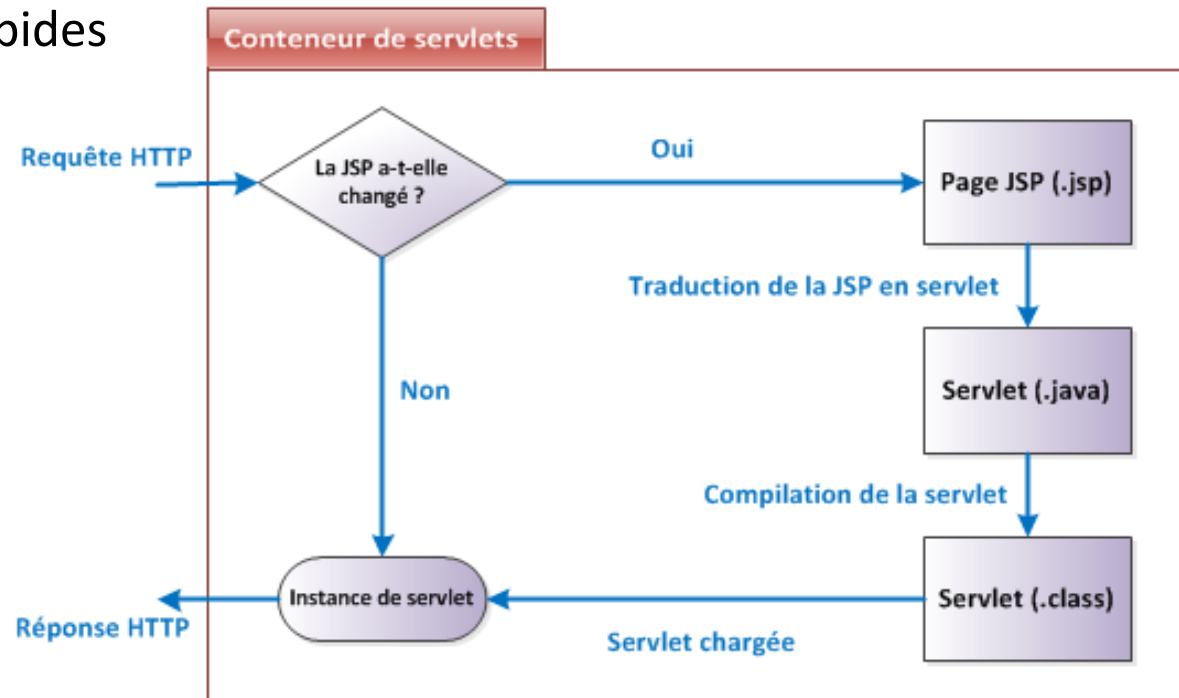
- La Servlet doit appeler la vue avec les méthodes *getRequestDispatcher* et *forward*
- Syntax:
`getServletContext().getRequestDispatcher("<maJSP>").forward(request, response);`
- La JSP a accès aux objets `request` et `response`
- Exemple

```
public void doGet( HttpServletRequest request, HttpServletResponse response ) throws Exception {  
    this.getServletContext().getRequestDispatcher( "/WEB-INF/test.jsp" ).forward( request,  
response );  
}
```

I. La technologie JSP

A savoir

- Les JSP sont compilées par un compilateur JSP
- Deviennent des **Servlets Java**
- La servlet ainsi générée et compilée est sauvegardée
- Les appels suivant sont plus rapides



I. La technologie JSP

Une JSP se compose

- D'une partie statique
 - Données invariables
 - Se limite en général à du code HTML
 - Une JSP sans code JAVA est valide
- D'une partie dynamique
 - Données générées selon les demandes du client
 - Introduites sous forme de balises

I. La technologie JSP

Exemple de JSP: Partie statique

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Test</title>
  </head>

  <body>
    <p>Ceci est une page générée depuis une JSP.</p>
  </body>
</html>
```

I. La technologie JSP

Plusieurs variables sont accessibles au sein d'une JSP

- `Out`
Objet utilisé pour envoyer la réponse HTTP au client
- `request`
Objet représentant la requête HTTP
- `response`
Objet représentant la réponse HTTP
- `Session`
Objet utilisé pour conserver de l'information
- Bien d'autres:
`page`, `pageContext`, `application`, etc.

A utiliser dans des balises !

Sommaire

1. La technologie JSP
- 2. Balises**
3. Directives
4. Transmission de données

II. Balises

Balise de commentaires: `<%-- ... %>`

- Permet d'insérer des commentaires dans le code JSP
`<%-- mon commentaires --%>`
- Les commentaires sont ignorés par le serveur
- Ne sont pas envoyés au client
- N'apparaissent pas dans le code source

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <%-- Ma premiere JSP --%>
    <h1>Hello world !</h1>
  </body>
</html>
```

II. Balises

Balise de déclaration: `<%! ... %>`

- Déclarer variables au sein d'une JSP
`<%! String t = "Bonjour"; %>`
- La variable est accessible partout dans la JSP
- Possibilité de déclarer plusieurs variables au sein d'un même bloc
- Possibilité de déclarer des fonctions dans le bloc
- **A EVITER! Le rôle de la vue n'est pas de faire du traitement !**

```
<%!  
String t = "Bonjour";  
  
public max(int a, int b){  
    return a>b?a:b;  
}  
%>
```

II. Balises

Balise de script: `<% ... %>`

- Permet d'insérer du code JAVA
- **A EVITER! Le rôle de la vue n'est pas de faire du traitement !**
- Exemple: affichage d'une table de multiplication

```
<%  
    for(int i = 1; i < 11; i++){  
        out.println("<h1 Table de "+i);  
        for(j = 1 ; j < 11; j++){  
            out.println(i + " X " + j + " = " + (i*j));  
        }  
        out.println("<hr />");  
    }  
%>
```

II. Balises

Balise d'expressions: `<%= ... %>`

- Offre des raccourcis d'écriture
- Renvoie le contenu d'une chaîne de caractères vers la réponse HTTP
- Plutôt que décrire:

`<% out.println(" Bonjour "); %>`

Ecrire

`<%= "Bonjour" %>`



- **Sans point virgule !**

Sommaire

1. La technologie JSP
2. Balises
- 3. Directives**
4. Transmission de données

III. Directives

Les directives JSP

- Contrôle la façon dont le compilateur crée la servlet associée
- Elle permettent
 - D'importer des packages
 - D'inclure d'autres JSP
 - Définir certaines propriétés
 - Inclure des bibliothèques de balises qui étendent les fonctionnalités
 - Placé entre les balises
`<%@ ... %>`
- Placé en entête de la JSP,
a l'exception de la directive d'inclusion de page

III. Directives

Directive de page

- Définit des informations relatives à al JSP
- Par exemple: importer des classes:
`<%@ import="java.util.List, java.util.Date" %>`
- De nombreuses autres possibilités: contentType, langue, session, etc.

```
<%@ page
    language="..."
    session="true | false"
    contentType="..."
    ...
%>
```

III. Directives

Directive d'inclusion

- Il est courant de découper une page WEB en plusieurs blocs
Ex. Entete, menu, pied, etc.
- L'inclusion permet d'inclure une JSP dans une autre
`<%@ include file="maJSP.jsp" %>`
- Identique à un copier-coller

Sommaire

1. La technologie JSP
2. Balises
3. Directives
- 4. Transmission de données**

IV. Transmission de données

On a vu que:

- La requête est dans un premier temps reçu par la servlet
- La servlet effectue un premier traitement
- Elle fait ensuite appel à la JSP

Pour ajouter de la dynamique:

- La servlet doit pouvoir transmettre des données à la JSP
- La JSP doit pouvoir récupérer ces données et les afficher

IV. Transmission de données

Transmettre des données

- Méthode `setAttribute` du coté de la Servlet
- Méthode `getAttribute` du coté de la JSP
- Tous les objets peuvent être transmis!
 - Un entier, un étudiant, une liste, un Object, etc...
- L'objet récupéré à l'aide de `getAttribute` doit être casté !

IV. Transmission de données

Exemple de transmission de données :

Servlet.java

```
class Servlet extends HttpServlet{
    public void doGet(HttpServletRequest req,
        HttpServletResponse res){
        String message = "A transmettre à la JSP" ;
        req.setAttribute("mess" , message);
        this.getServletContext().getRequestDispatcher(
            "WEB/INF/mapJSP.jsp").forward(req, res);
    }
}
```

maJSP.jsp

```
<!DOCTYPE html>
<html>
    <head></head>
    <body>
        <%
            String t = (String)request.getAttribute("mess");
            out.println(t);
        %>
    </body>
</html>
```

IV. Transmission de données

Transmettre des données

- La JSP a accès aux objets `request` et `response`
- La JSP peut accéder directement aux paramètres avec la méthode `getParameter`

IV. Transmission de données

Une JSP peut rediriger l'utilisateur

- Idem que pour Servlet
- Pour une redirection en interne méthode

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <%
      request.getRequestDispatcher("/page").forward(request, response); %>
    %>
  </body>
</html>
```