

COURS JAVA EE ET SOA

Présenté par

Moustapha DER

Enseignant – Chercheur – Chef de projet – CEO

AUTEUR DU LIVRE : « **LE GUIDE DU GENIE LOGICIEL** »

Contacts : 77 656 84 51 / 77 836 12 12 **Email** : djamilder@gmail.com

PRESENTATION JSP

- **La** technologie **JSP** est utilisée pour créer une application Web, tout comme la technologie Servlet. Il peut être considéré comme une extension de Servlet car il fournit plus de fonctionnalités qu'un servlet tel que le langage d'expression, JSTL, etc.
- Une page JSP se compose de balises HTML et de balises JSP. Les pages JSP sont plus faciles à maintenir que Servlet car nous pouvons séparer la conception et le développement. Il fournit des fonctionnalités supplémentaires telles que le langage d'expression, les balises personnalisées, etc.

AVANTAGES DE JSP SUR SERVLET

- Il existe de nombreux avantages de JSP par rapport au servlet. Ils sont les suivants:
- 1) Extension au servlet
- La technologie JSP est l'extension de la technologie Servlet. Nous pouvons utiliser toutes les fonctionnalités du servlet dans JSP. De plus, nous pouvons utiliser des objets implicites, des balises prédéfinies, un langage d'expression et des balises personnalisées dans JSP, ce qui facilite le développement de JSP.
- 2) Facile à entretenir
- JSP peut être facilement géré car nous pouvons facilement séparer notre logique métier avec la logique de présentation. Dans la technologie Servlet, nous combinons notre logique métier avec la logique de présentation.

AVANTAGES DE JSP SUR SERVLET

- 3) Développement rapide: pas besoin de recompiler et de redéployer
- Si la page JSP est modifiée, nous n'avons pas besoin de recompiler et de redéployer le projet. Le code du servlet doit être mis à jour et recompilé si nous devons changer l'aspect et la convivialité de l'application.
- 4) Moins de code que Servlet
- Dans JSP, nous pouvons utiliser de nombreuses balises telles que des balises d'action, JSTL, des balises personnalisées, etc. qui réduisent le code. De plus, nous pouvons utiliser EL, des objets implicites, etc.

LE CYCLE DE VIE D'UNE PAGE JSP

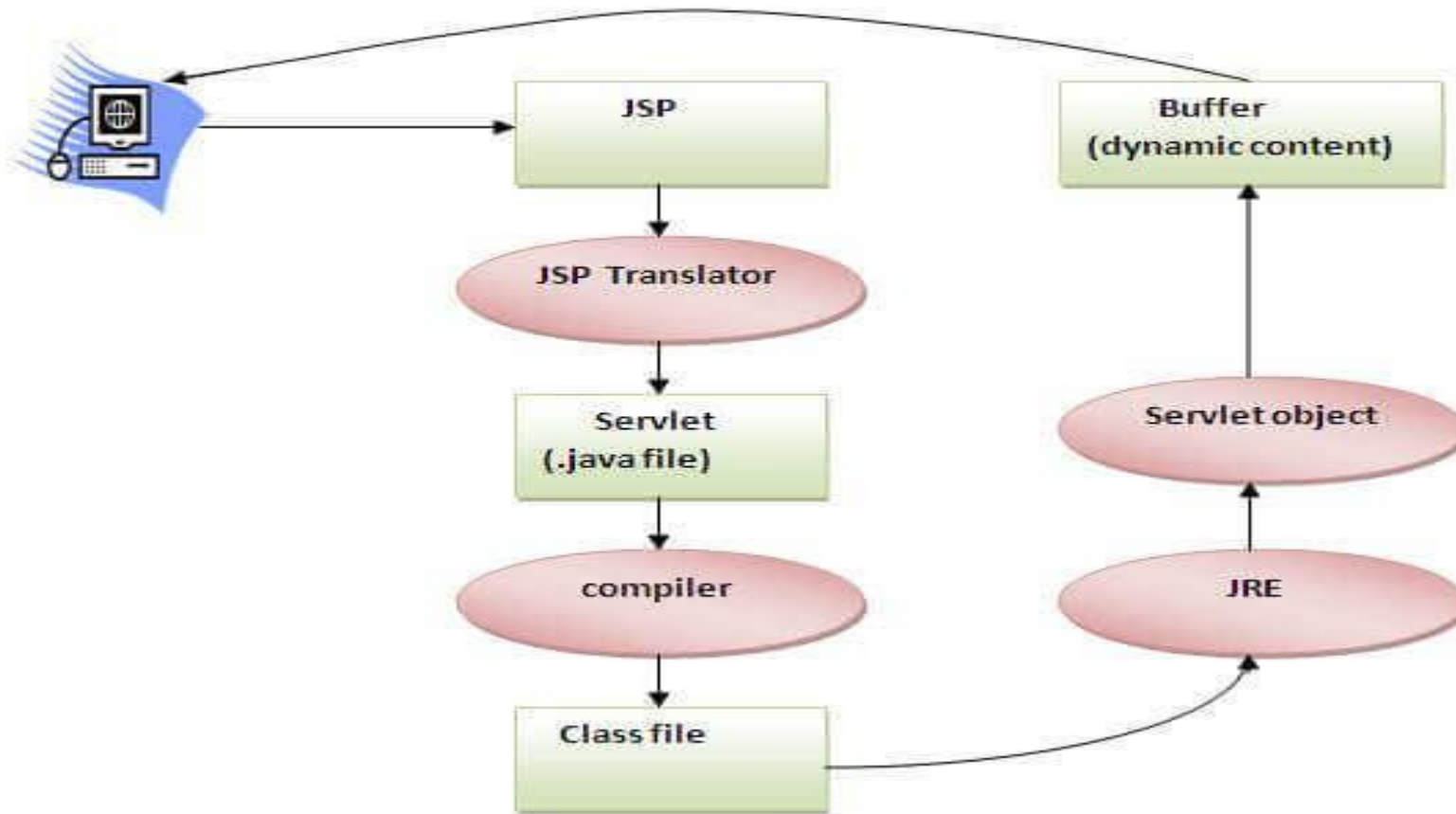
- Les pages JSP suivent ces phases:
- Traduction de la page JSP
- Compilation de la page JSP
- Chargement de classe (le chargeur de classe charge le fichier de classe)
- Instanciation (l'objet du servlet généré est créé).
- Initialisation (le conteneur appelle la méthode `jspInit ()`).
- Traitement de la demande (le conteneur appelle la méthode `_jspService ()`).
- Destroy (le conteneur appelle la méthode `jspDestroy ()`).

LE CYCLE DE VIE D'UNE PAGE JSP

- Les pages JSP suivent ces phases:
- Traduction de la page JSP
- Compilation de la page JSP
- Chargement de classe (le chargeur de classe charge le fichier de classe)
- Instanciation (l'objet du servlet généré est créé).
- Initialisation (le conteneur appelle la méthode `jspInit ()`).
- Traitement de la demande (le conteneur appelle la méthode `_jspService ()`).
- Destroy (le conteneur appelle la méthode `jspDestroy ()`).

Remarque: `jspInit ()`, `_jspService ()` et `jspDestroy ()` sont les méthodes de cycle de vie de JSP.

LE CYCLE DE VIE D'UNE PAGE JSP



Comme illustré dans le diagramme ci-dessus, la page JSP est traduite en Servlet à l'aide du traducteur JSP. Le traducteur JSP fait partie du serveur Web qui est responsable de la traduction de la page JSP en servlet. Après cela, la page Servlet est compilée par le compilateur et convertie en fichier de classe. De plus, tous les processus qui se produisent dans Servlet sont exécutés sur JSP plus tard, comme l'initialisation, la validation de la réponse au navigateur et la destruction.

BALISE JSP SCRIPTLET (ÉLÉMENTS DE SCRIPT)

- Dans JSP, le code java peut être écrit à l'intérieur de la page jsp à l'aide de la balise scriptlet. Voyons d'abord quels sont les éléments de script.
- **Éléments de script JSP**
- Les éléments de script offrent la possibilité d'insérer du code java dans le jsp. Il existe trois types d'éléments de script:
- balise de scriptlet
- balise d'expression
- balise de déclaration

TP

Présenté par Moustapha DER Enseignant – Chercheur – Chef de projet – CEO - AUTEUR DU LIVRE : « **LE GUIDE DU GENIE LOGICIEL** » **Contacts** : 77 656 84 51 / 77 836 12 12 **Email** : djamilder@gmail.com

BALISE DE SCRIPTLET JSP

- Une balise de scriptlet est utilisée pour exécuter le code source Java dans JSP. La syntaxe est la suivante: `<% code source java%>`

1. `<html>`

2. `<body>`

3. `<% out.print("LES ETUDIANTS DE DAR SONT DES GENIES"); %>`

4. `</body>`

5. `</html>`

TP

Présenté par Moustapha DER Enseignant – Chercheur – Chef de projet – CEO - AUTEUR DU LIVRE : « **LE GUIDE DU GENIE LOGICIEL** » **Contacts** : 77 656 84 51 / 77 836 12 12 **Email** : djamilder@gmail.com

BALISE D'EXPRESSION JSP

- Le code placé dans la **balise d'expression JSP** est *écrit dans le flux de sortie de la réponse* . Vous n'avez donc pas besoin d'écrire `out.print ()` pour écrire des données. Il est principalement utilisé pour imprimer les valeurs de variable ou de méthode.
- Syntaxe de la balise d'expression JSP
- `< % = instruction% >`

EXEMPLE DE BALISE D'EXPRESSION JSP

- Dans cet exemple de balise d'expression jsp, nous affichons simplement un message de bienvenue.

1.< **html** >

2.< **body** >

3.< % = "bienvenue EN DAR"% >

4.</ **body** >

5.</ **html** >

Remarque: ne terminez pas votre instruction par un point-virgule en cas de balise d'expression.

TP

Présenté par Moustapha DER Enseignant – Chercheur – Chef de projet – CEO - AUTEUR DU LIVRE : « **LE GUIDE DU GENIE LOGICIEL** » **Contacts** : 77 656 84 51 / 77 836 12 12 **Email** : djamilder@gmail.com

EXEMPLE DE BALISE D'EXPRESSION JSP QUI IMPRIME L'HEURE ACTUELLE

- Pour afficher l'heure actuelle, nous avons utilisé la méthode `getTime ()` de la classe `Calendar`. `GetTime ()` est une méthode d'instance de la classe `Calendar`, nous l'avons donc appelée après avoir récupéré l'instance de la classe `Calendar` par la méthode `getInstance ()`.

1.<html>

2.<body>

3.HEURE

COURANTE: <%= java.util.Calendar.getInstance().getTime() %>

4.</body>

5.</html>

BALISE DE DÉCLARATION JSP

- La **balise de déclaration JSP** est utilisée *pour déclarer des champs et des méthodes* .
- Le code écrit à l'intérieur de la balise de déclaration jsp est placé en dehors de la méthode service () du servlet généré automatiquement.
- Donc, il ne reçoit pas de mémoire à chaque demande.

SYNTAXE DE LA BALISE DE DÉCLARATION JSP

- La syntaxe de la balise de déclaration est la suivante:

< %! déclaration de champ ou de
méthode% >

DIFFÉRENCE ENTRE LA BALISE DE SCRIPTLET JSP ET LA BALISE DE DÉCLARATION

Balise de scriptlet Jsp	Balise de déclaration Jsp
La balise de scriptlet jsp ne peut déclarer que des variables et non des méthodes.	La balise de déclaration jsp peut déclarer des variables ainsi que des méthodes.
La déclaration de la balise scriptlet est placée dans la méthode <code>_jspService ()</code> .	La déclaration de la balise de déclaration jsp est placée en dehors de la méthode <code>_jspService ()</code> .

EXEMPLE DE BALISE DE DÉCLARATION JSP QUI DÉCLARE UN CHAMP

- Dans cet exemple de balise de déclaration JSP, nous déclarons le champ et imprimons la valeur du champ déclaré à l'aide de la balise d'expression jsp.

1.<html>

2.<body>

3.<%! int n=50; %>

4.<%= "Value of the variable is:"+n %>

5.</body>

6.</html>

EXEMPLE DE BALISE DE DÉCLARATION JSP QUI DÉCLARE LA MÉTHODE

- Dans cet exemple de balise de déclaration JSP, nous définissons la méthode qui renvoie le cube de nombre donné et appelons cette méthode à partir de la balise d'expression jsp. Mais nous pouvons également utiliser la balise scriptlet jsp pour appeler la méthode déclarée.

1.<html>

2.<body>

3.<%!

4.int cube(int n){

5.return n*n*n*;

6.}

7.%>

8.<%= "Cube of 3 is:"+cube(3) %>

9.</body>

10.</html>

BALISES D'ACTION JSP

- Il existe de nombreuses balises ou éléments d'action JSP. Chaque balise d'action JSP est utilisée pour effectuer certaines tâches spécifiques.
- Les balises d'action sont utilisées pour contrôler le flux entre les pages et pour utiliser Java Bean. Les balises d'action Jsp sont indiquées ci-dessous.

BALISES D'ACTION JSP

Balises d'action JSP	Description
jsp:forward	transfère la demande et la réponse à une autre ressource.
jsp: include	comprend une autre ressource.
jsp: useBean	crée ou localise un objet bean.
jsp: setProperty	définit la valeur de la propriété dans l'objet bean.
jsp: getProperty	imprime la valeur de propriété du bean.
jsp: plugin	incorpore d'autres composants tels que l'applet.
jsp:param	définit la valeur du paramètre. Il est utilisé en avant et comprend principalement.
jsp:fallback	peut être utilisé pour imprimer le message si le plugin fonctionne. Il est utilisé dans jsp: plugin.

JSP: BALISE D'ACTION AVANT

- La balise d'action jsp: forward est utilisée pour transmettre la demande à une autre ressource, que ce soit jsp, html ou une autre ressource.
- Syntaxe de jsp: balise d'action avant sans paramètre
 1. <jsp:forward page="relativeURL | <%= expression %>" />
- Syntaxe de jsp: balise d'action avant avec paramètre
 1. <jsp:forward page="relativeURL | <%= expression %>">
 2. <jsp:param name="parametername" value="parametervalue | <%= expression %>" />
 3. </jsp:forward>

EXEMPLE DE JSP: BALISE D'ACTION AVANT AVEC PARAMÈTRE

1. Test.jsp

```
2.<html>
3.<body>
4.<h2>this is index page</h2>
5. <jsp:forward page="EXO.jsp" />
6.</body>
7.</html>
```

1. EXO.jsp

```
2.<html>
3.<body>
4.<% out.print("Today is:" + java.util.Calendar.getInstance().getTime()); %>
5.</body>
6.</html>
```

EXEMPLE DE JSP: BALISE D'ACTION AVANT AVEC PARAMÈTRE

- Dans cet exemple, nous transmettons la demande au fichier `aven` passant les paramètres au fichier `EXO.jsp` qui affiche la valeur du paramètre avec la date et l'heure.

```
1.<html>
2.<body>
3.<h2> c'est la page d'index </h2>
4. <jsp: forward page = "EXO.jsp" >
5.<jsp: param name = "nom" value = "esmt.sn" />
6.</ jsp: forward >
7. </body>
8.</html>
```

EXEMPLE DE JSP: BALISE D'ACTION AVANT AVEC PARAMÈTRE

1.EXO.jsp

2.<html>

3.<body>

4. <% out.print ("Aujourd'hui est:" + java.util.Calendar.getInstance ().
getTime ()); %>

5.<% = request.getParameter ("nom")%>

6. </body>

7.</html>

TP

Présenté par Moustapha DER Enseignant – Chercheur – Chef de projet – CEO - AUTEUR DU LIVRE : « **LE GUIDE DU GENIE LOGICIEL** » **Contacts** : 77 656 84 51 / 77 836 12 12 **Email** : djamilder@gmail.com

JSP: INCLUDE UNE BALISE D'ACTION

- La **balise d'action jsp: include** est utilisée pour inclure le contenu d'une autre ressource, qu'il s'agisse de jsp, html ou servlet.
- La balise d'action jsp include inclut la ressource au moment de la demande, elle est donc **préférable pour les pages dynamiques**, car des modifications pourraient être apportées à l'avenir.
- La balise jsp: include peut être utilisée pour inclure des pages statiques et dynamiques.
- **Avantage de jsp: inclure une balise d'action**
- **Réutilisation du code** : nous pouvons utiliser une page plusieurs fois, par exemple en incluant des pages d'en-tête et de pied de page dans toutes les pages. Cela fait donc gagner beaucoup de temps.

SYNTAXE DE JSP

- Syntaxe de jsp: inclure une balise d'action sans paramètre
 - 1.<jsp: include page = "relativeURL | <% = expression%>" />
- Syntaxe de jsp: inclure une balise d'action avec un paramètre
 - 1.<jsp: include page = "relativeURL | <% = expression%>" >
 - 2.<jsp: param name = "parametername" value = "parametervalue | <% = expression%>" />
 - 3.</ jsp: include>

EXEMPLE DE JSP: INCLURE UNE BALISE D'ACTION SANS PARAMÈTRE

- Dans cet exemple, le fichier index.jsp inclut le contenu du fichier printdate.jsp.

- *Fichier: test.jsp*

1. <h2> **c'est la** page d'index </h2>
2. <jsp: include page = "affdate.jsp" />
3. <h2> section de fin de la page d'index </h2>

- *Fichier: affdate.jsp*

1. <% out.print ("Aujourd'hui est:" + java.util.Calendar.getInstance (). getTime ()); %>

JAVABEAN

- Un JavaBean est une classe Java qui doit suivre les conventions suivantes:
 - Il devrait avoir un constructeur sans argument.
 - Il devrait être sérialisable.
 - Il doit fournir des méthodes pour définir et obtenir les valeurs des propriétés, appelées méthodes getter et setter.
- Pourquoi utiliser JavaBean?
- C'est un composant logiciel réutilisable. Un bean encapsule de nombreux objets dans un seul objet afin que nous puissions accéder à cet objet à partir de plusieurs endroits. De plus, il permet un entretien facile.

EXEMPLE

```
1.//Employee.java
2. package mypack;
3.public class Employee implements java.io.Serializable{
4.private int id;
5.private String name;
6.public Employee(){ }
7.public void setId(int id){this.id=id;}
8.public int getId(){return id;}
9.public void setName(String name){this.name=name;}
10.public String getName(){return name;}
11.}
```

COMMENT ACCÉDER À LA CLASSE JAVABEAN?

POUR ACCÉDER À LA CLASSE JAVABEAN, NOUS DEVONS UTILISER LES MÉTHODES GETTER ET SETTER.

```
1.package mypack;  
2.public class Test{  
3.public static void main(String args[]){  
4.Employee e=new Employee();//object is created  
5.e.setName("M. DER");//setting value to the object  
6.System.out.println(e.getName());  
7.}  
8.}
```

TP

Présenté par Moustapha DER Enseignant – Chercheur – Chef de projet – CEO - AUTEUR DU LIVRE : « **LE GUIDE DU GENIE LOGICIEL** » **Contacts** : 77 656 84 51 / 77 836 12 12 **Email** : djamilder@gmail.com