

# Le développement Web côté serveur – BackEnd avec Java EE

## Module 6 – Les JSP avancées



1

Les JSP avancées

## Objectifs

- Connaître les principales technologies disponibles
- Savoir utiliser les principales balises JSP
- Savoir utiliser l'Expression Language (EL)
- Savoir utiliser les principales bibliothèques JSTL



2

## La principale limite des JSP

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
...
<body>
<%
if(session.getAttribute("compteurAcces")!=null){
    int compteurAcces = (int)session.getAttribute("compteurAcces");
%>
    <p>Nombre d'accès : <%=compteurAcces%></p>
<%}
else{
%>
    <p>Le compteur d'accès n'existe pas en session</p>
<%}%>
<p>
    Avec le lien suivant, le suivi de session n'est assuré que si les cookies sont autorisés :
    <a href="<%=request.getContextPath()%>/modules/module5/ServletManipulationSession">
        ServletManipulationSession (sans réécriture d'URL)</a>
</p>
...
</body>
</html>
```

*Trop de Java*

*Fichier  
potentiellement  
grand*

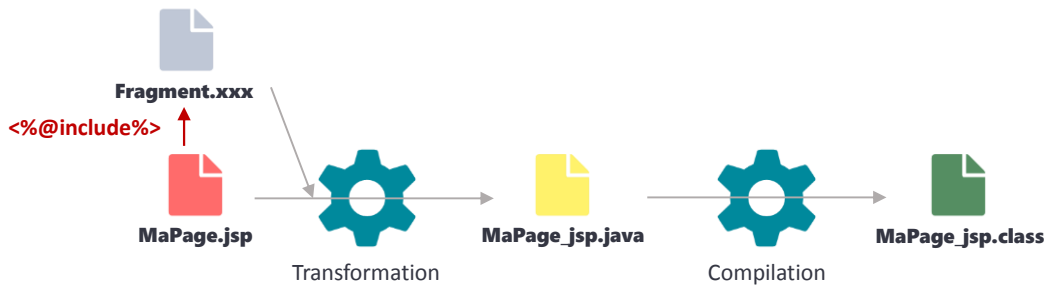


## Les principales technologies disponibles

- La directive `<%@include%>`
- Les balises JSP
- L'Expression Language (EL)
- Les bibliothèques JSTL



## La directive `<%@include%>`



## Les balises JSP

- Elles cachent du code Java en le remplaçant

```
<jsp:xxx/>
```

- Inclusions

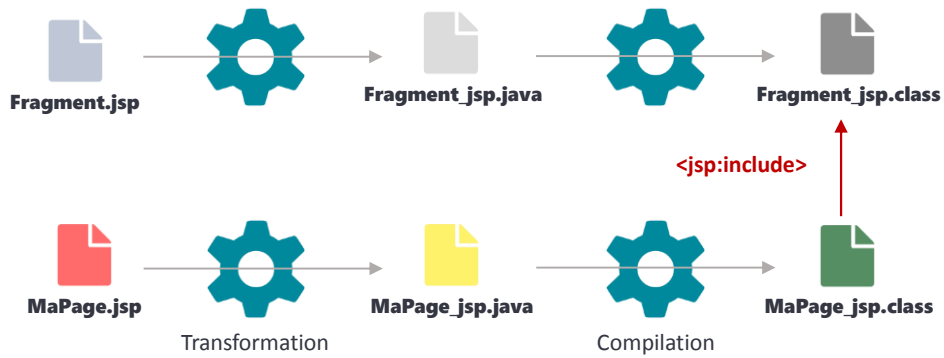
```
<jsp:include page="url de la page">  
  <jsp:param value="valeur" name="parametre"/>  
</jsp:include>
```

- Lecture de données

*Abordée juste  
après...*



## La balise `<jsp:include>`



## Les inclusions dans une JSP

### Démonstration



## Les balises <jsp:useBean/> et <jsp:getProperty/>

- Lecture de données

```
<jsp:useBean
    id="nomBean"
    scope="request"
    class="nom.package.NomClasse">

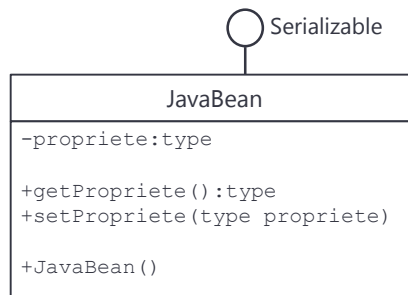
</jsp:useBean>

<jsp:getProperty
    property="proprieteDuBean"
    name="nomBean"/>
```



## La notion de JavaBean

- Classe Java respectant des règles utiles à de nombreux frameworks



La classe n'est pas **final**



Les JSP avancées

Les balises `<jsp:useBean/>` et `<jsp:getProperty/>`

## Démonstration



11

Les JSP avancées

La spécification EL

# Expression Language 3.0



JSR 341



12

## La syntaxe de l'EL

- Syntaxe simple  
`${expression}`
- Nomenclature pointée  
`${monObjet.maVariable}`  
`${monObjet.monTableau[0]}`  
`${monObjet.maMethode() }`
- Structure de code impossible
  - Pas d'alternative
  - Par de boucle
- Ne lève jamais d'exception



## Lecture d'attributs avec l'EL

- Lecture ciblée dans un contexte  
`${pageScope['nomAttribut']}` **ou** `${pageScope.nomAttribut}`  
`${requestScope['nomAttribut']}` **ou** `${requestScope.nomAttribut}`  
`${sessionScope['nomAttribut']}` **ou** `${sessionScope.nomAttribut}`  
`${applicationScope['nomAttribut']}` **ou** `${applicationScope.nomAttribut}`
- Lecture non ciblée  
`${nomAttribut}`
  - Recherche dans l'ordre suivant : page, request, session et application



## Les principaux autres objets disponibles

- Pour la lecture des paramètres de requête  
`${param.nomDuParametre}`
- Pour la lecture des en-têtes de requête  
`${header.nomEntete}`
- Pour la lecture des cookies  
`${cookie.nomCookie}`
- Pour la lecture du contexte de page  
`${pageContext.request.contextPath}`



## L'Expression Language (EL)

### Démonstration





# Java Server Pages Standard Tag Library 1.2



JSR 52



17

| Nom de la bibliothèque | Rôle  |
|------------------------|---|
| core                   | Librairie de base<br>Outillage pour alternatives, répétitives...        |
| x                      | Librairie de manipulation XML   |
| fmt                    | Librairie d'internationalisation et de formatage                        |
| sql                    | Librairie pour faire des traitements SQL                                |
| functions              | Librairie de fonctions utilitaires disponibles pour les instructions EL |



18

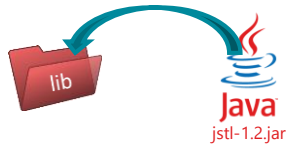
## La directive `<%@ taglib %>`

- Permet l'import d'une librairie JSTL
- Syntaxe

```
<%@ taglib prefix="XXX" uri="namespace de la librairie" %>
```

- Utilisation de la librairie

```
<XXX:balise attribut="valeur"></XXX:balise>
```



## La librairie core

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

- Les alternatives

```
<c:if test="${expression EL}">  
  ...  
</c:if>
```

```
<c:choose>  
  <c:when test="${expression EL}">  
    ...  
  </c:when>  
  <c:otherwise>  
    ...  
  </c:otherwise>  
</c:choose>
```



## La librairie core

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

- Les répétitives

```
<c:forEach var="element" items="${collection}">  
    ${element.propriete}  
</c:forEach>
```



## Quelques opérateurs utiles de l'EL

| Domaine                        | Opérateurs disponibles                                |
|--------------------------------|---|
| Les opérateurs de comparaison  | == (eq), != (ne), <(lt), >(gt), <=(le), >=(ge), empty |
| Les opérateurs logiques        | &&(and),   (or), !(not)                               |
| L'opérateur conditionnel       | ? :   |
| Les opérateurs arithmétiques   | +, -, *, /(div), %(mod)                               |
| ⚠ L'opérateur de concaténation | +=  |

```
<c:if test="${maVariable==12}">  
<c:if test="${!empty maVariable}">
```



## La librairie fonctions

```
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
```

- Pas de balise
- Utilisation dans les expressions EL

```
{fn:}
```

- contains - boolean contains(java.lang.String, java.lang.String)
- containsIgnoreCase - boolean containsIgnoreCase(java.lang.String, java.lang.String)
- endsWith - boolean endsWith(java.lang.String, java.lang.String)
- escapeXml - java.lang.String escapeXml(java.lang.String)
- indexOf - int indexOf(java.lang.String, java.lang.String)
- join - java.lang.String join(java.lang.String[], java.lang.String)
- length - int length(java.lang.Object)
- replace - java.lang.String replace(java.lang.String, java.lang.String, java.lang.String)
- split - java.lang.String[] split(java.lang.String, java.lang.String)
- startsWith - boolean startsWith(java.lang.String, java.lang.String)
- substring - java.lang.String substring(java.lang.String, int, int)
- substringAfter - java.lang.String substringAfter(java.lang.String, java.lang.String)
- substringBefore - java.lang.String substringBefore(java.lang.String, java.lang.String)
- toLowerCase - java.lang.String toLowerCase(java.lang.String)
- toUpperCase - java.lang.String toUpperCase(java.lang.String)
- trim - java.lang.String trim(java.lang.String)

Press 'Ctrl+Space' to show JSP Proposals



## Les balises JSTL

# Démonstration



# Gérer vos listes de courses

TP



## Conclusion

- Vous connaissez des technologies complémentaires pour améliorer vos JSP

