

Le développement Web côté serveur avec Java EE

Module 3 – Les JSP



Objectifs

- Comprendre le rôle d'une JSP
- Savoir positionner la JSP dans une architecture MVC
- Savoir réaliser une JSP simple
- Comprendre le fonctionnement d'une JSP
- Savoir faire communiquer une servlet et une JSP
- Savoir écrire du code Java dans une JSP
- Savoir gérer proprement les erreurs

Les JSP

La spécification

JavaServer Pages 2.3



JSR 245

Les JSP

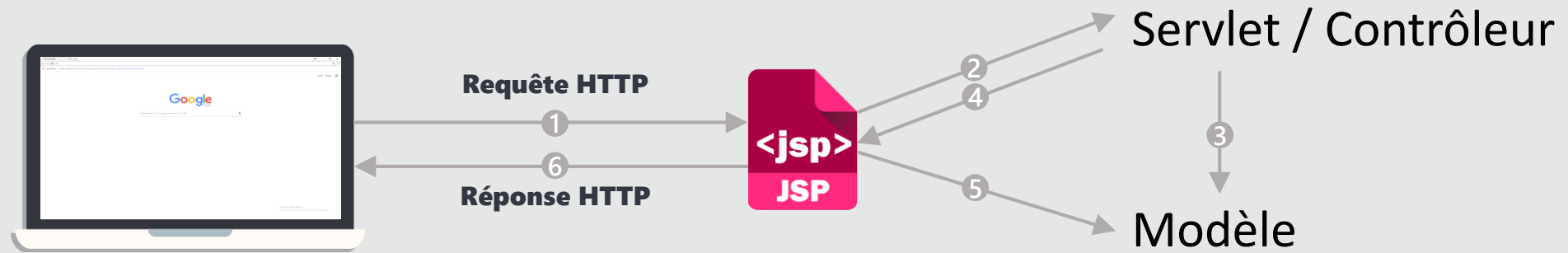
Qu'est-ce que c'est ?

La vue!

Pas de
traitement métier!

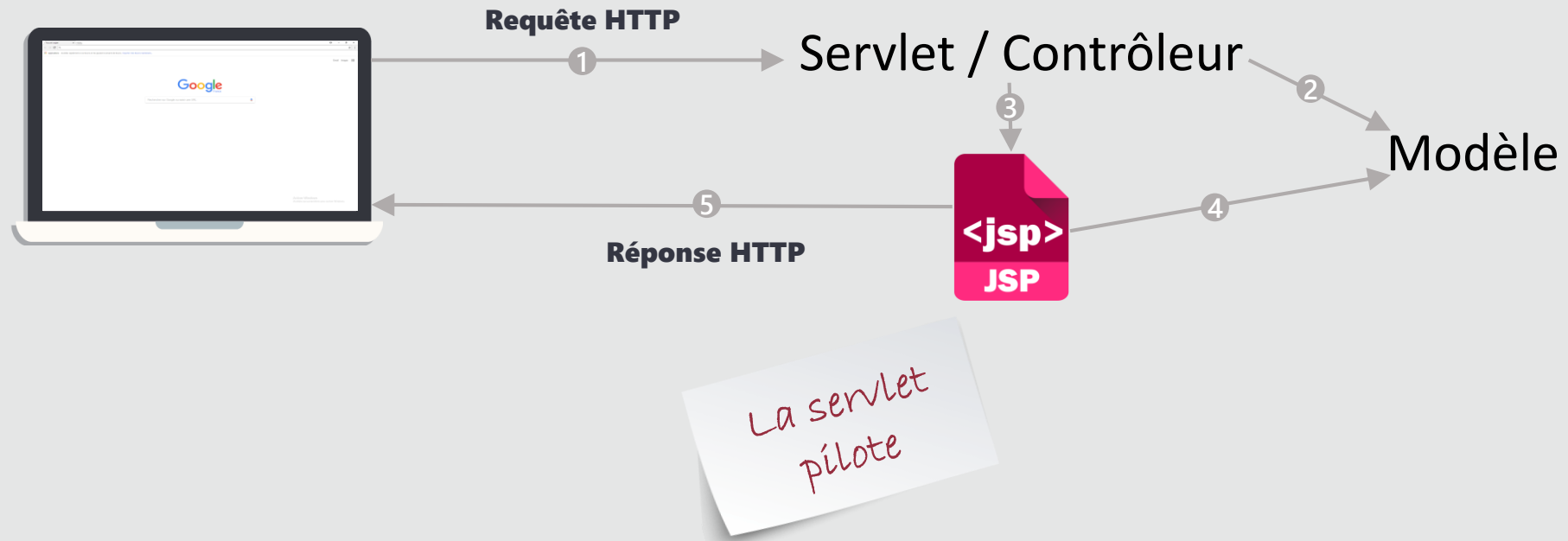
uniquement
l'affichage

Intégration dans une architecture MVC



La JSP pilote

Intégration dans une architecture MVC2



La création d'une page JSP

```
<%@ page language="java"
      contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>

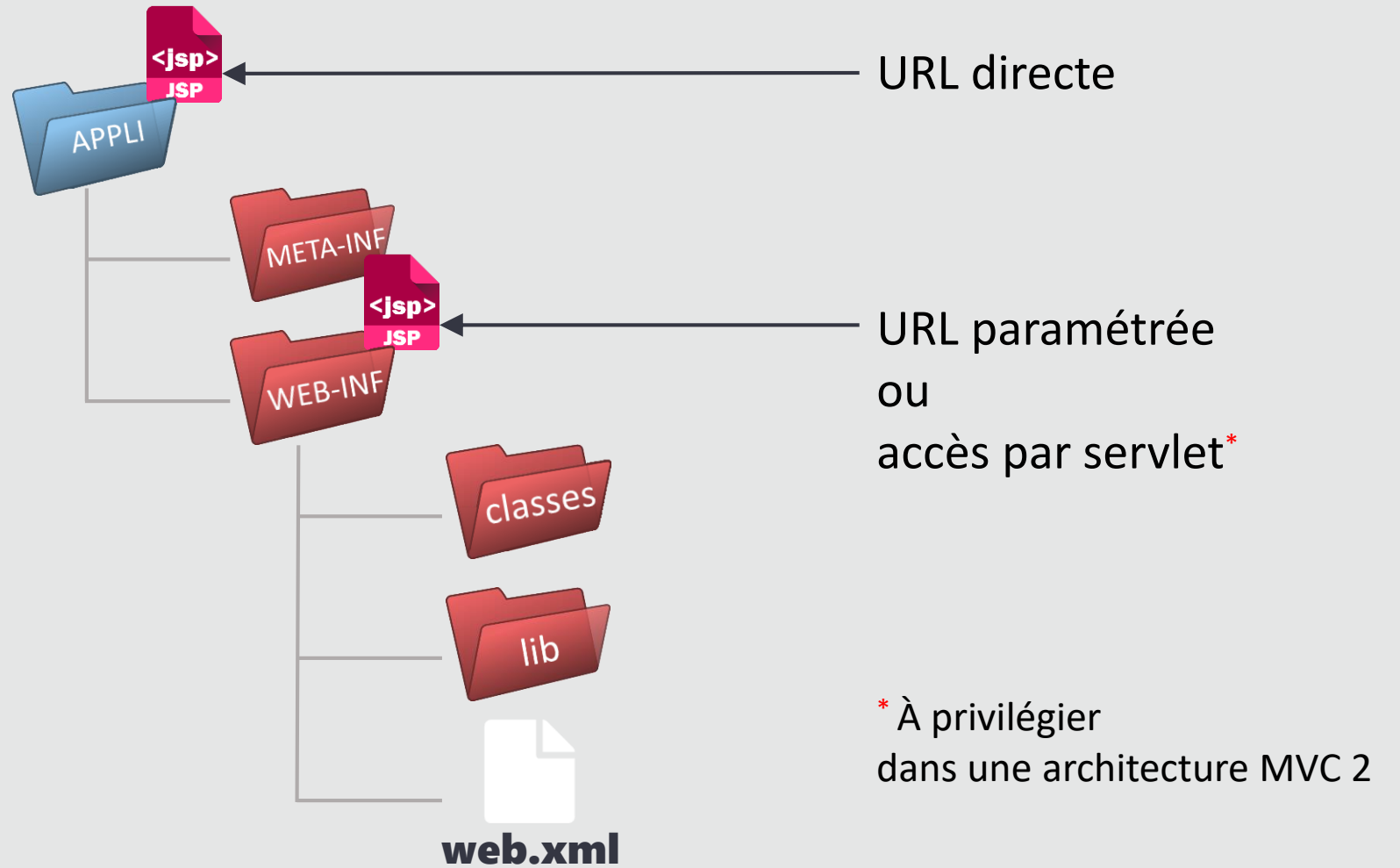
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Titre de la JSP</title>
</head>
<body>

</body>
</html>
```

- Une directive page
- Un contenu HTML



Les URL d'accès ?



Le paramétrage de l'URL



web.xml

```
<!-- Déclaration d'une JSP -->
<servlet>
    <servlet-name>uneJSP</servlet-name>
    <jsp-file>/WEB-INF/uneJSP.jsp</jsp-file>
</servlet>

<!-- Association de la JSP à une ou des URL -->
<servlet-mapping>
    <servlet-name>uneJSP</servlet-name>
    <url-pattern>/url/de/la/jsp</url-pattern>
</servlet-mapping>
```

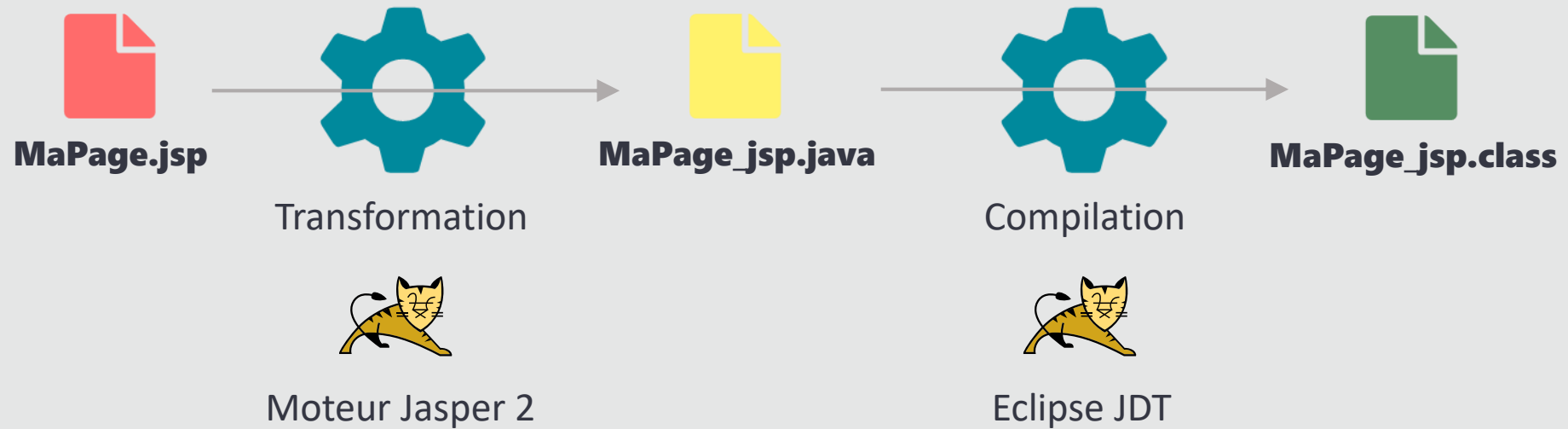
Les JSP

La première JSP

Démonstration

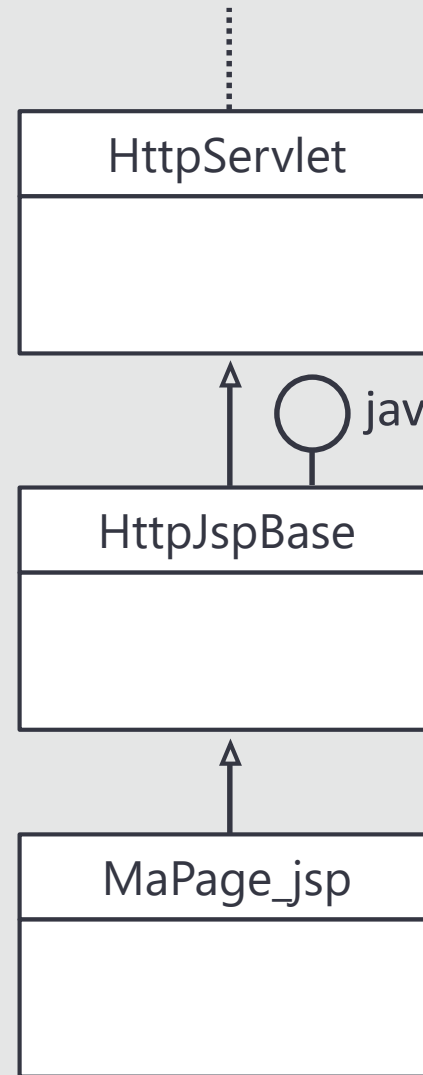


Le principe d'exécution



Modèle objet

*une JSP
n'est qu'une
servlet*



javax.servlet.http



javax.servlet.jsp.HttpJspPage



org.apache.jasper.runtime



org.apache.jsp

Le cycle de vie d'une JSP

*Le même
que
les servlets*

UneServlet

```
init()  
destroy()  
doXXX()
```

MaPage_jsp

```
_jspInit()  
_jspDestroy()  
_jspService()
```

Un exemple

```
<%@ page
    language="java"
    contentType="text/html;
    charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"
%>
<html>
...
</html>
```



```
package org.apache.jsp.modules.module3.demonstrations;

public final class premiereJSP_jsp
    extends org.apache.jasper.runtime.HttpJspBase
{
    //...
    public void _jspInit() {}
    public void _jspDestroy() {}

    public void _jspService(
        final javax.servlet.http.HttpServletRequest request,
        final javax.servlet.http.HttpServletResponse response)
        throws java.io.IOException, javax.servlet.ServletException {
        //...
        out.write("\r\n");
        out.write("<html>\r\n");
        //...
        out.write("</html>");
        //...
    }
}
```

Les JSP

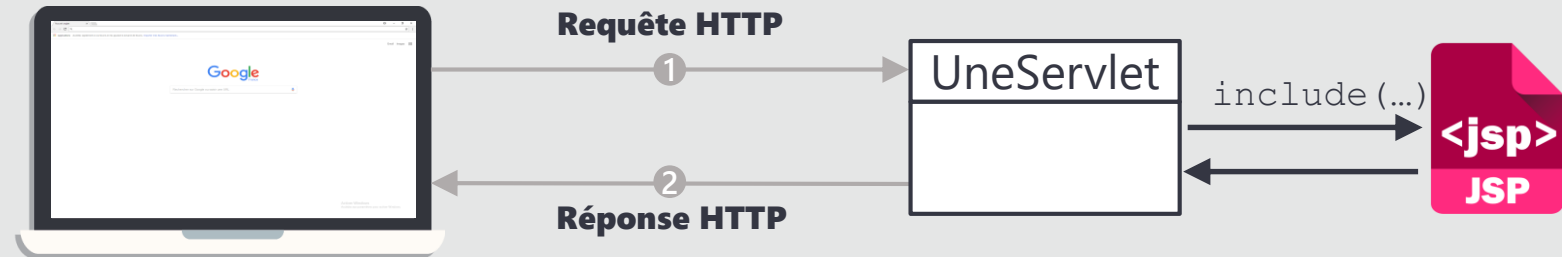
Le principe d'exécution

Démonstration

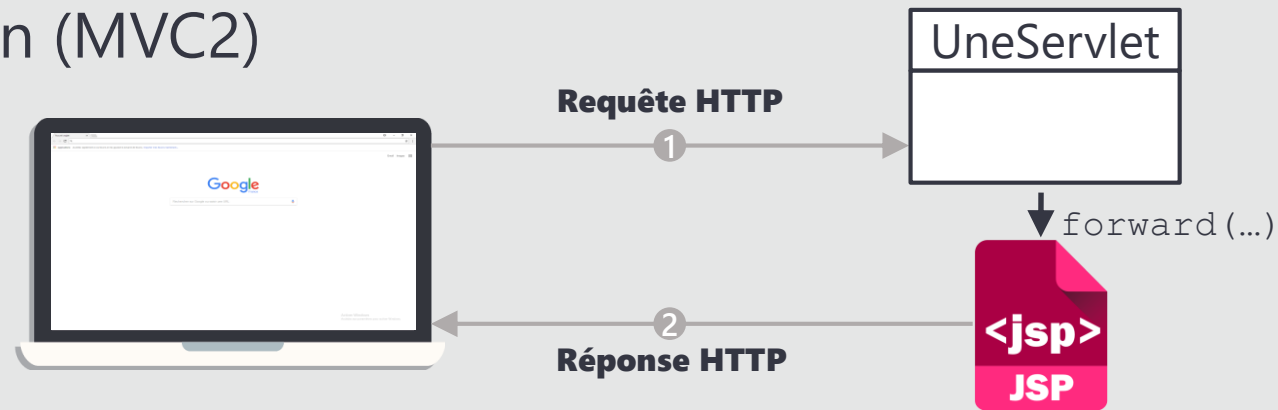


L'appel depuis une servlet

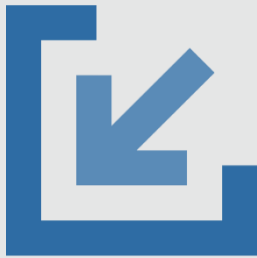
- Par inclusion



- Par délégation (MVC2)



Les méthodes en jeu



RequestDispatcher
<<interface>>

```
include(HttpServletRequest request,  
        HttpServletResponse response)
```

```
forward(HttpServletRequest request,  
        HttpServletResponse response)
```



Exemple d'inclusion et de délégation

//Récupération du disptacheur

RequestDispatcher rd;

rd = request.getRequestDispatcher("URL");

rd = this.getServletContext().getRequestDispatcher("URL");

rd = this.getServletContext().getNamedDispatcher("Nom de la ressource");

JSP ou autre

//traitement...

//inclusion

rd.include(request, response);

//suite traitement

//traitement

//forward

rd.forward(request, response);

⚠ //terminé

Les JSP

La délégation

Démonstration



Ecrire du code Java dans les JSP

- Les éléments de script
- Les objets Java disponibles
- Les directives

```
<%[!|=]  
code java  
%>
```

```
<%@ une directive %>
```

Les éléments de script

```
public class MaPage_jsp extends ... {
```

- Les déclarations

<%!

code java

%>

```
public void _jspService(...)
```

- Les scriptlets

<%

code java

%>

```
{
```

```
...
```

```
...
```

```
...
```

```
out.print(...);
```

```
...
```

```
}
```

```
}
```

- Les expressions

<%=information à afficher*%>



* Pas de point-virgule

Un exemple

```
<%@ page language="java"
      contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>

<%!
    private int nombreAppels;
    //+getter/setter

%>
<html>
...
<%
    this.setNombreAppels(this.getNombreAppels()+1);

%>

Nombre d'appels de la JSP :
<input type="text"
      value="<%=this.getNombreAppels() %>"/>
...
```



```
public final class presentationElementsDeScript_jsp
    extends org.apache.jasper.runtime.HttpJspBase
{
    private int nombreAppels;
    //+getter/setter

    public void _jspService(...) {
        //...
        out.write("<html>\r\n");
        //...
        this.setNombreAppels(this.getNombreAppels()+1);
        //...
        out.print(this.getNombreAppels());
        //...
    }
}
```

Les JSP

Les éléments de script

Démonstration



Les principaux objets disponibles

- request

```
<a href="<%=request.getContextPath() %>/chemin vers la ressource">  
    lien depuis la racine de l'application  
</a>
```

- response

```
<%  
    response.setHeader("un en-tête", "une valeur");  
%>
```

- out

```
<%  
    out.print("une information");  
%>
```


Les JSP

Les principaux objets disponibles

Démonstration



La communication entre la servlet et la JSP

- Dans la servlet

```
request.setAttribute("cle", "valeur de n'importe quel type");
```

- Dans la JSP

```
type maVariable = (type)request.getAttribute("cle");
```

- La durée de vie est limitée à la requête

Les JSP

La communication entre la servlet et la JSP

Démonstration



Les JSP
Chifoumi

TP



La directive page

```
<%@ page language="java"
contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"
errorPage="nom de la page d'erreur associée"
isErrorPage="true|false"%>
```

```
<%@ page import="nom de la classe" %>
```

La gestion des erreurs dans les JSP

- Page pouvant générer des erreurs

```
<%@ page language="java"  
errorPage="nom de la page d'erreur associée"  
isErrorPage="false"%>
```

- Page traitant les erreurs

```
<%@ page language="java"  
isErrorPage="true"%>
```

- Exploitation de l'erreur

```
<%=exception.getMessage () %>
```

Les JSP

La gestion des erreurs dans les JSP

Démonstration



Les JSP

Chifoumi (évolution)

TP



Conclusion

- Vous savez maintenant générer une réponse adaptée avec la technologie des JSP