

Plugin TILES

Introduction :

Le plugin Tiles permet de gérer l'apparence et le découpage des applications web.

Les projets Internet utilisent habituellement les découpages sous forme d'entête, de menu, de contenu et de pied de page. Pour réaliser ce découpage, les développeurs utilisent des tableaux HTML ou mieux, des définitions à base de feuilles de styles CSS et de balises ``, `<div/>` ou autres.

En effet, si la présentation change (le nom des fragments) ou la mise en page, nous devons modifier chaque page utilisatrice.

Pour éviter cela, le plugin Tiles fournit une librairie de tags permettant de définir la mise en page pour toutes les pages utilisatrices du site. Un changement dans la définition des pages entraîne une modification instantanée de toutes les pages utilisatrices. Le plugin Tiles repose sur deux composants, le **modèle** et la **définition**. Le modèle des pages est Défini à partir d'une page JSP .Chaque JSP utilisée par le modèle de découpage sera déclarée dans la page. Maintenant si nous voulons changer tout le modèle de présentation de l'application Il suffit de changer une seule page JSP de définition.

1-Page de mise en forme du modèle :

La page JSP ci dessous permet de définir le modèle de présentation du site.

```

<%@ taglib uri="http://tiles.apache.org/tags-tiles" prefix="tiles"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title><tiles:getAsString name="titrepage"/></title>
</head>
<body>
  <table border="1" cellpadding="2" cellspacing="2" align="center">
    <tr>
      <td height="30" colspan="2"><tiles:insertAttribute name="header" />
      </td>
    </tr>
    <tr>
      <td height="250"><tiles:insertAttribute name="menu" /></td>
      <td width="350"><tiles:insertAttribute name="body" /></td>
    </tr>
    <tr>
      <td height="30" colspan="2"><tiles:insertAttribute name="footer" />
      </td>
    </tr>
  </table>
</body>
</html>

```

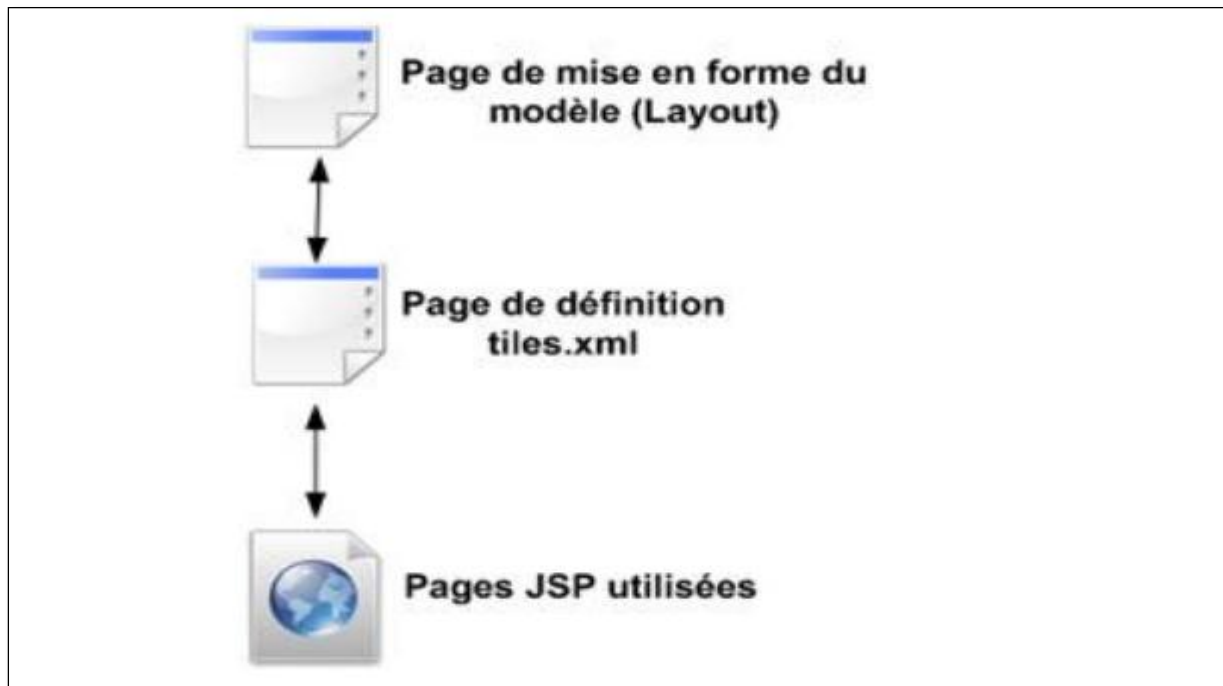
Nous retrouvons en début de fichier l'inclusion de la librairie de tags et deux utilisations de balises :

<tiles:getAsString name="titrepage"/> : Permet de retourner une chaîne de caractère définie dans le fichier de configuration.

<tiles:insertAttribute name="entete"/> : Permet d'inclure une page dans la définition du modèle de présentation.

2. Définition du modèle :

Une définition de modèle est utilisée entre la page de présentation du modèle et les pages JSP utilisatrices. Par analogie Java, la page de présentation du modèle (layout) est souvent comparée à une interface et la définition du modèle à la classe utilisatrice de l'interface fournissant une implémentation de celle-ci.



[Tiles.xml :](#)

La page de définition du modèle est précisée dans un fichier *tiles.xml* présent dans le répertoire */WEBINF* de l'application. Le fichier *tiles.xml* ci-dessous est présenté à l'aide de la balise `<definition/>` et des attributs `name` et `template`.

```

<?xml version="1.0" encoding="UTF-8" ?>

  <!DOCTYPE tiles-definitions PUBLIC
    "-//Apache Software Foundation//DTD Tiles Configuration 2.0//EN"
    "http://tiles.apache.org/dtds/tiles-config_2_0.dtd">

  <tiles-definitions>

    <definition name="Layout" template="/Layout.jsp">
      <put-attribute name="titrepage" value="Template" />
      <put-attribute name="header" value="/header.jsp" />
      <put-attribute name="menu" value="/menu.jsp" />
      <put-attribute name="body" value="/body.jsp" />
      <put-attribute name="footer" value="/footer.jsp" />
    </definition>

    <definition name="welcome" extends="Layout">
      <put-attribute name="titrepage" value="Welcome" />
      <put-attribute name="body" value="/welcome.jsp" />
    </definition>

    <definition name="page1" extends="Layout">
      <put-attribute name="titrepage" value="page1" />
      <put-attribute name="body" value="/page1.jsp" />
    </definition>

    <definition name="page2" extends="Layout">
      <put-attribute name="titrepage" value="page2" />
      <put-attribute name="body" value="/page2.jsp" />
    </definition>

  </tiles-definitions>

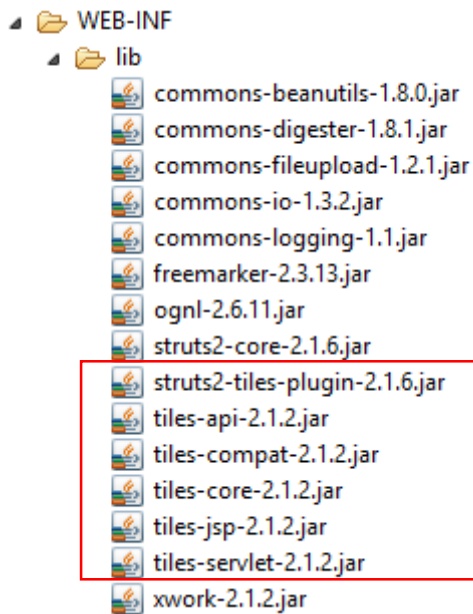
```

La balise <definition/> contient une ou plusieurs balises <put-attribute/> utilisées en référence à la page de présentation du modèle. Pour notre exemple la page de présentation utilisée par le modèle est /Layout.jsp. On a déclaré une variable nommée titrepage associée à la présentation du modèle et trois pages afin d'insérer respectivement l'en-tête, le contenu et le pied de page. Par la suite, dans le fichier de configuration de l'application struts.xml, les résultats de type tiles seront associés aux différentes définitions du fichier tiles.xml (welcome, page1 et page2).

3. Mise en place du plug-in Tiles :

La mise en place du plug-in Tiles est réalisée en quatre étapes :

- ✚ Copie des archives suivants dans le répertoire /WEB-INF/lib de l'application.



- ✚ Définition du listener associé à Tiles dans le fichier de configuration /WEB-INF/web.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  id="WebApp_ID" version="2.5">
  <display-name>Struts2Example15</display-name>

  <context-param>
    <param-name>org.apache.tiles.impl.BasicTilesContainer.DEFINITIONS_CONFIG</param-name>
    <param-value>/WEB-INF/tiles.xml</param-value>
  </context-param>

  <listener>
    <listener-class>org.apache.struts2.tiles.StrutsTilesListener</listener-class>
  </listener>

  <filter>
    <filter-name>struts2</filter-name>
    <filter-class>
      org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
  </filter>

  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

- ✚ Définition des résultats de type tiles dans le fichier de configuration struts.xml.

```
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
  <package name="default" extends="struts-default">
    <result-types>
      <result-type name="tiles"
        class="org.apache.struts2.views.tiles.TilesResult" />
    </result-types>
    <action name="*Link" method="{1}" class="com.mql.struts2.tiles.TilesAction">
      <result name="welcome" type="tiles">welcome</result>
      <result name="page1" type="tiles">page1</result>
      <result name="page2" type="tiles">page2</result>
    </action>
  </package>
</struts>
```

- ✚ Utilisation des résultats de types tiles dans les résultats des actions Struts.

Le fichier struts.xml contient la définition des actions et des résultats de type tiles. Ces derniers sont en rapport avec le fichier de définition du modèle Layout.jsp et les définitions du fichier tiles.xml.

Les résultats définis dans le fichier struts.xml sont envoyés au modèle de présentation Layout.jsp et utilisent les données présentent dans les définitions welcome, page1 et page2 du fichier tiles.xml.

Test de l'exemple :

Lorsqu'on exécute notre exemple on se trouve avec la page « welcome.jsp » qui va nous montrer le découpage sous forme d'entête, de menu, de contenu et de pied de page.

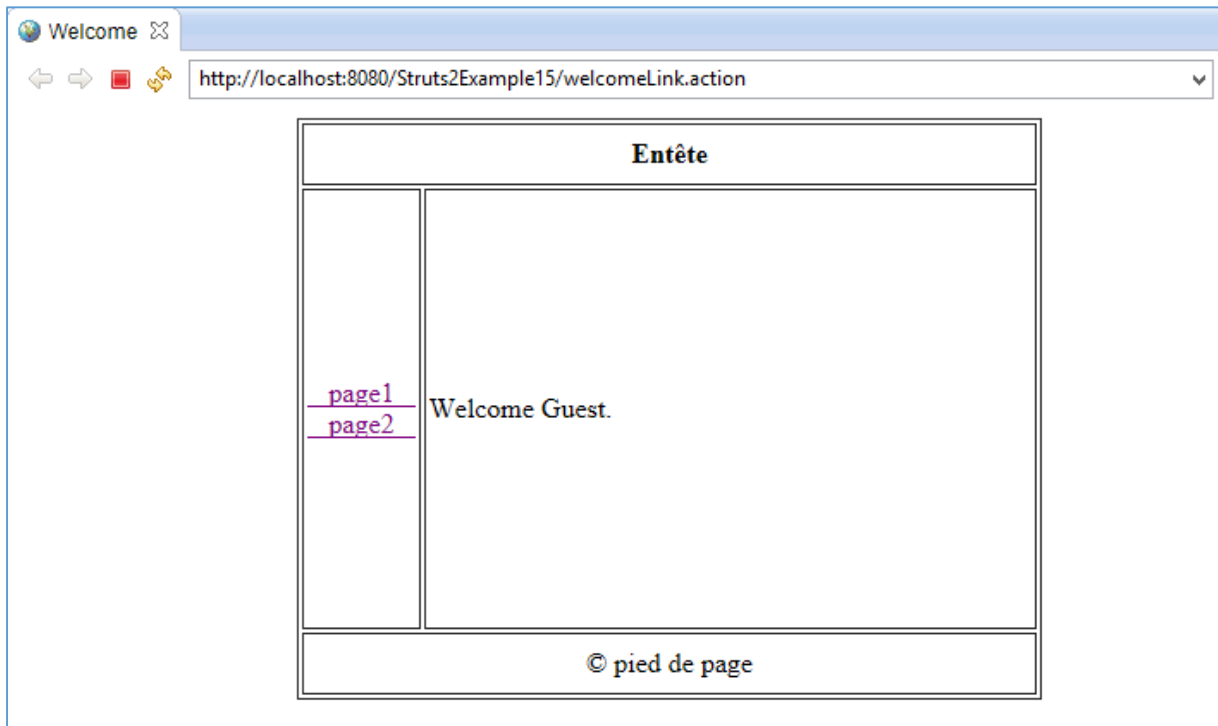


Figure 1 : Page d'accueil « Welcome.jsp »

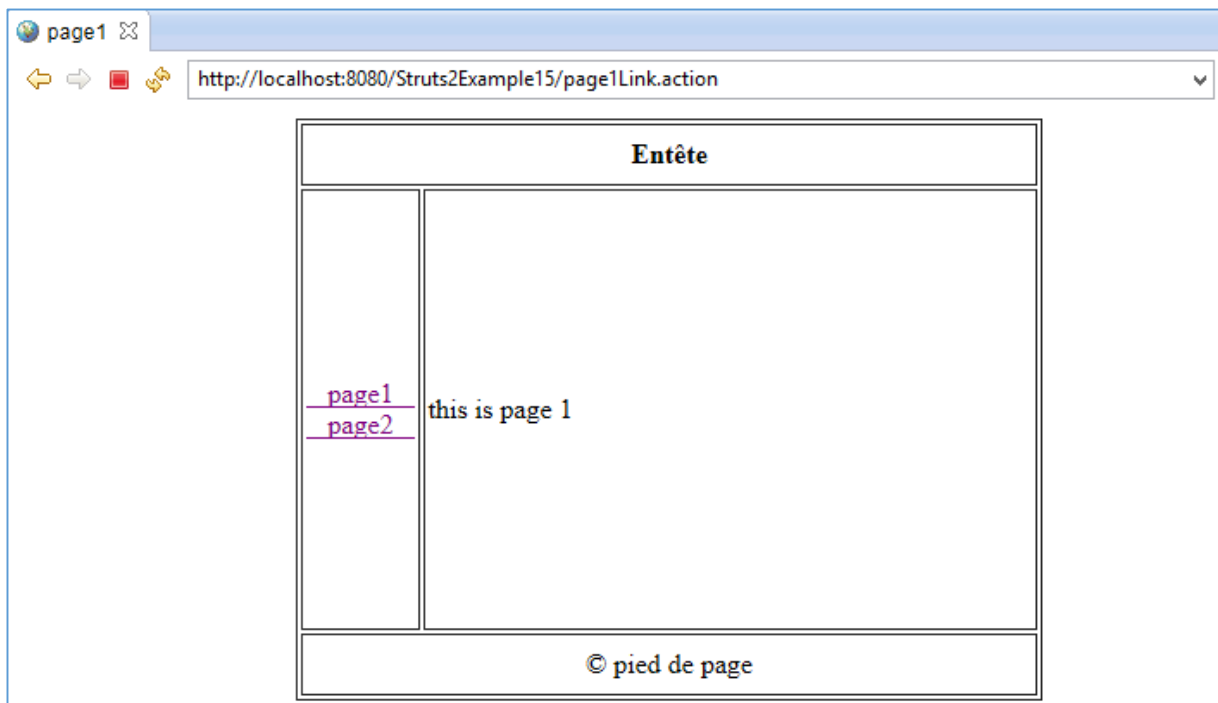


Figure 2 : « page1.jsp »

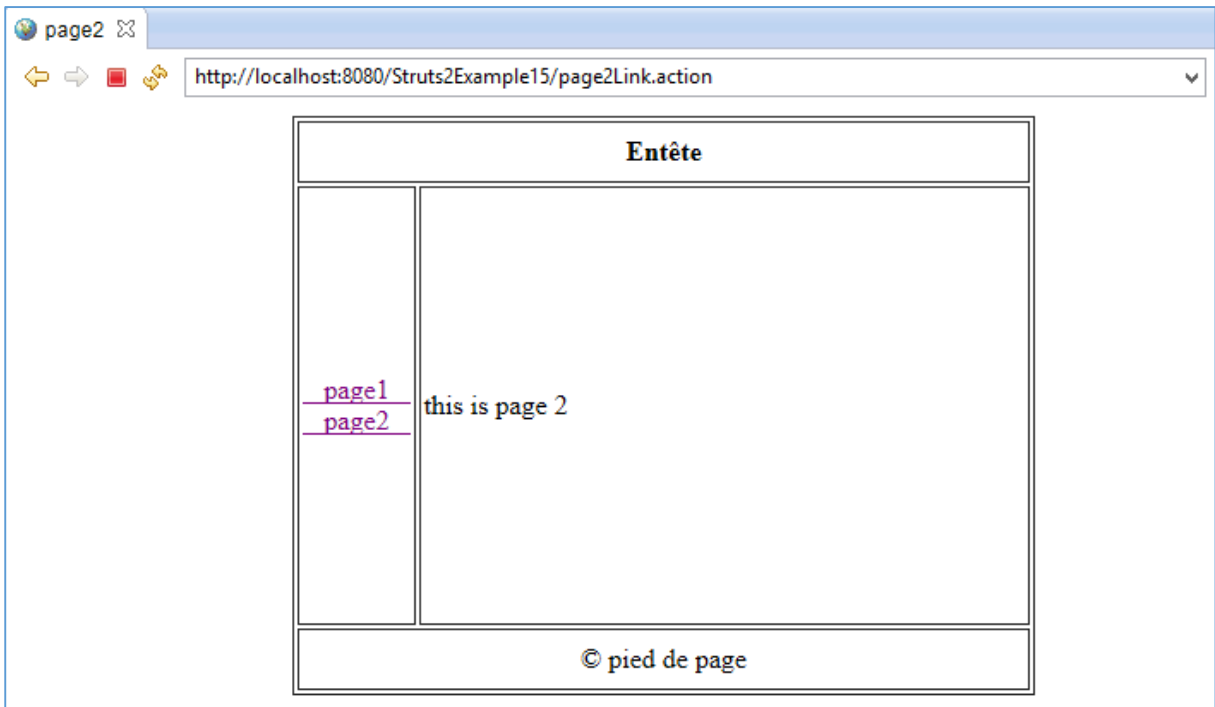


Figure 3 : « page2.jsp »