



Framework Corporatiu J2EE

Servei de Presentació:Pantalles

Versió 1.1

Barcelona, 2 / octubre / 2006



Històric de modificacions

Data	Autor	Comentaris	Versió
09/12/2005	Atos Origin, sae openTrends	Versió inicial del document	1.0
02/10/2006	Atos Origin	Versió 1.1 openFrame	1.1

Llegenda de Marcadors



Índex

1.	INTRODUCCIÓ	4
1.1.	PROPÓSIT	4
1.2.	CONTEXT I ESCENARIS D'ÚS	4
1.3.	VERSIONS I DEPENDÈNCIES	5
1.3.1.	<i>Versions</i>	5
1.3.2.	<i>Dependències Bàsiques</i>	5
1.4.	A QUI VA DIRIGIT	6
1.5.	DOCUMENTS I FONTS DE REFERÈNCIA	6
1.6.	GLOSSARI	6
2.	DESCRIPCIÓ DETALLADA	7
2.1.	ARQUITECTURA I COMPONENTS	7
2.2.	INSTAL·LACIÓ I CONFIGURACIÓ	7
2.2.1.	<i>Instal·lació</i>	7
2.2.2.	<i>Configuració</i>	7
2.3.	UTILITZACIÓ DEL SERVEI	8
2.3.1.	<i>Definició de les Pantalles</i>	8
2.3.2.	<i>Definició de les Pàgines Plantilla</i>	8
2.3.3.	<i>Retorn de les Pàgines des de les classes Action</i>	8
2.4.	EINES DE SUPORT	8
2.5.	INTEGRACIÓ AMB ALTRES SERVEIS	8
2.6.	PREGUNTES FREQUÈNTS	8
3.	EXEMPLES	9
3.1.	EXEMPLE DE PLANTILLA GENÉRICA	9
3.1.1.	<i>Definició de la Pàgina</i>	9
4.	ANNEXOS	11
4.1.	TILES	11
4.1.1.	<i>Definició d'una Pantalla basada en un Template</i>	11



1. Introducció

1.1. Propòsit

En tota aplicació Web es necessita un Look&Feel (L&F) comú. Si el L&F és "hardcoded" en totes les pàgines un canvi de L&F es converteix en un autèntic malson, ja que hauríem de modificar totes les pàgines de l'aplicació. En un bon disseny és important que aconseguim una separació clara entre el L&F i el contingut de les pàgines.

L'ús de plantilles ens permet definir un L&F comú, on podem especificar que usarem una capçalera, un cos, un menú a l'esquerra i un peu. La pàgina final es construeix passant-li com a paràmetre quina capçalera, quin cos i quin menú correspon a la pantalla a mostrar com a resposta a la petició.

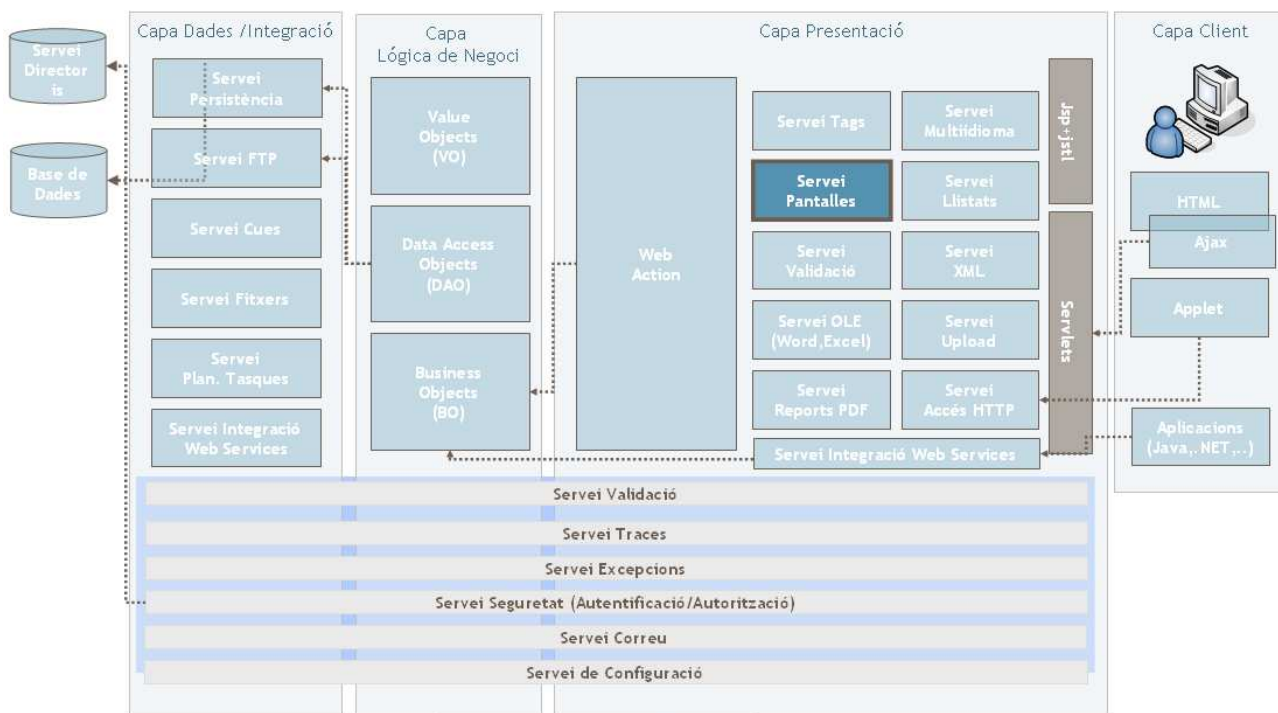
1.2. Context i Escenaris d'Ús

El Servei de Pantalles es troba associat a l'ús de Java Server Pages com a tecnologia de presentació dinàmica de contingut Web basat en HTML. En la tecnologia JSP (Java Server Pages) existeixen 2 formes estàndar d'incloure contingut d'altres pàgines: de forma estàtica i de forma dinàmica.

- **Estàtica:** `<%@ include file="..." %>`
Si usem `<%@ include file="..." %>`, s'inclou el contingut de la pàgina destí en temps de **compilació** i no d'execució.
D'aquesta manera, el text de la pàgina inclosa s'insereix en la del pare. Si la pàgina inclosa és una pàgina JSP (podria ser un HTML, ...), s'avalua i el seu contingut s'insereix en la pàgina pare. Després, la pàgina pare continua traduint la següent línia.
- **Dinàmica:** `<jsp:include page="..." />`
Si usem `<jsp:include page="..." />` el recurs és tractat dinàmicament, de manera que en **temps d'execució** s'avalua la pàgina i es volca el resultat.

L'ús d'aquests mecanismes és utilitzat en alguns àmbits com la solució a la incorporació de parts específiques d'una pantalla. Tot i ser una solució àmpliament utilitzada no permet poder especificar plantilles reutilitzables en la nostra aplicació. El Servei de Pantalles en contraposició ens permet definir herència de pantalles i definir estructures comuns reutilitzables en la nostra aplicació.

El Servei de Pantalles es troba dins dels serveis de Presentació de openFrame, tal i com es mostra en el següent gràfic:



La implementació del servei es basa en la utilització de Spring amb Struts i Tiles. Per tant, es considera prerequisite el coneixement de la utilització de Tiles.

1.3. Versions i Dependències

1.3.1. Versions

No s'han produït canvis respecte la versió 1.0.

1.3.2. Dependències Bàsiques

En el present apartat es mostren quines són les versions i dependències necessàries per fer ús del Servei.

Per a la utilització del Servei de Pantalles de openFrame es requereix d'un entorn basat en les següents tecnologies:

- JSP
- Struts

Nom	Tipus	Versió	Descripció
-----	-------	--------	------------



Nom	Tipus	Versió	Descripció
openFrame-services-web	jar	1.1	
servlet-api	jar	2.4	
struts	jar	1.2.7	

Per cada servei openFrame fer ús també de les dependències requerides per ell.

1.4. A qui va dirigit

Aquest document va dirigit als següents perfils:

- Programador. Per conèixer l'ús del servei
- Arquitecte. Per conèixer quins són els components i la configuració del servei

1.5. Documents i Fonts de Referència

Documentació
Tiles

<http://struts.apache.org/struts-tiles/index.html>

1.6. Glossari



2. Descripció Detallada

2.1. Arquitectura i Components

L'arquitectura del Servei de Pantalles es basa en la utilització de Struts Tiles. Per a més informació consultar '<http://struts.apache.org/struts-tiles/index.html>'.

2.2. Instal·lació i Configuració

Prerequisit: Per al correcte funcionament d'aquest servei és important que prèviament s'hagin realitzat les configuracions especificades a l'apartat 'Configuració Bàsica' del document 'Serveis de Presentació'.

2.2.1. Instal·lació

Per a la instal·lació del Servei de Pantalles es requereix de l'ús del fitxer 'openFrame-services-web.jar' i les dependències indicades a l'apartat 'Versions i Dependències'.

2.2.2. Configuració

La configuració del Servei de Pantalles es troba ja detallada al document general 'Serveis de Presentació' a l'apartat 'Configuració de Struts'. A mode de resum, direm que s'han d'haver realitzat els següents passos:

- 1) Definició del control·lador basat en Tiles

Fitxer de configuració: struts-config.xml

Ubicació proposada: <PROJECT_ROOT>/src/main/resources/struts/struts-config.xml

```
<controller>
  <set-property property="processorClass"
value="net.opentrends.openframe.services.web.struts.ExtendedDelegatingTilesRequest
Processor"/>
</controller>
```

- 2) Definició del Plugin de Tiles i la ubicació del fitxer de definicions de pantalles

Fitxer de configuració: struts-config.xml

Ubicació proposada: <PROJECT_ROOT>/src/main/resources/struts/struts-config.xml

```
<plug-in className="org.apache.struts.tiles.TilesPlugin">
  <set-property property="definitions-config" value="/WEB-
INF/classes/struts/tiles-definitions.xml"/>
</plug-in>

<plug-in className="org.springframework.web.struts.ContextLoaderPlugIn">
  <set-property property="contextConfigLocation" value="/WEB-
INF/classes/spring/action-servlet.xml"/>
```



```
</plug-in>
```

2.3. Utilització del Servei

El Servei es basa en la utilització de Tiles. Si es coneix com usar aquesta tecnologia ja serem capaços d'usar el Servei amb garanties. Es recomana una lectura prèvia de l'Annex .

2.3.1. Definició de les Pantalles

Consultar l'apartat '1.2 XML Definitions Configuration' de la url <http://struts.apache.org/struts-tiles/userGuide.html>.

2.3.2. Definició de les Pàgines Plantilla

Cadascuna de les pantalles es basen en un fitxer JSP que defineix el layout i les parts dinàmiques que seran substituïdes pels valors definits per les pantalles.

A les pàgines plantilla hem de fer ús de tags de Tiles, per això, en primer lloc, la pàgina ha d'importar la llibreria:

```
<%@ taglib uri="http://jakarta.apache.org/struts/tags-tiles" prefix="tiles"%>
```

La pàgina definida pot fer ús dels tags definits per Tiles. Veure apartats '1.3 i següents' de la url <http://struts.apache.org/struts-tiles/userGuide.html>.

2.3.3. Retorn de les Pàgines des de les classes Action

Una vegada definida una pàgina de l'aplicació amb el mecanisme de plantilles, el seu retorn es realitza des de l'atribut 'path' del fitxer de configuració de Struts.

2.4. Eines de Suport

2.5. Integració amb Altres Serveis

2.6. Preguntes Freqüents

3. Exemples

3.1. Exemple de Plantilla Genérica

En el següent exemple podem veure la definició de la plantilla genérica d'una aplicació:

```
<definition name="site.mainLayoutPetStore"
    path="/WEB-INF/jsp/layouts/mainLayout.jsp❶">
    <put name="title" value="Petstore Demo" />
    <put name="optionsBar" value="/WEB-INF/jsp/includes/optionsBar.jsp" />
    <put name="quickSearch" value="/WEB-INF/jsp/includes/quickHeader.jsp" />
    <put name="header" value="/WEB-INF/jsp/includes/top.jsp" />
    <put name="menu" value="/WEB-INF/jsp/includes/blank.jsp" />
    <put name="footer" value="/WEB-INF/jsp/includes/bottom.jsp" />
</definition>
```

- ❶ A l'atribut 'path' hem definit la ruta a la pàgina (veure a continuació 'Definició de la Pàgina') que conté l'estructura de les pàgines que es basaran en aquesta plantilla
- ❷ Mitjançant el tag 'put' hem indicat que substituïrem el lloc de la pàgina 'mainLayout.jsp' (segons l'atribut path) on aparegui el tag '`<tiles:insert attribute="nom"/>`' per la pàgina especificada a l'atribut 'value'

3.1.1. Definició de la Pàgina

```
<%@ page contentType="text/html; charset=ISO-8859-1"%>
<%
    response.setHeader("Cache-Control","no-cache"); //HTTP 1.1
    response.setHeader("Pragma","no-cache"); //HTTP 1.0
    response.setDateHeader("Expires", 0); //prevents caching at the proxy server
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%@ include file="/WEB-INF/jsp/includes/fwkTagLibs.jsp" %>
<html>
<head>
<title><tiles:insert attribute="title"/></title>
<link rel="stylesheet" href='<%=request.getContextPath()%>/css/openFrame.css'
type="text/css"/>
<fwk:configuration styleId="defaultConfiguration" />
<script>
    var contextPath = '<%=request.getContextPath()%>';
</script>
</head>
<body>
    <tiles:insert attribute="header"/>
    <TABLE BORDER="0" CELLPADDING="0" CELLSPACING="0" WIDTH="100%">
        <TR><TD ALIGN="CENTER" CLASS="COPY" style="border-top: #000000
1px solid; "> <tiles:insert attribute='optionsBar' /> </TD></TR>
    </TABLE>
    <TABLE BORDER="0" CELLPADDING="0" CELLSPACING="1" WIDTH="100%">
        <TR>
```



```
<TD ALIGN="CENTER" VALIGN="TOP"><BR>
    <table border="0" cellpadding="2">
    <tr>
        <td valign="top"><tiles:insert
attribute='menu' /></td>
        <TD WIDTH="100%" VALIGN="TOP">
        <!-- Contenido principal -->

        <!--TABLE bgcolor="#000000"
cellpadding="0" cellspacing="1" border="0" width="100%">
        <tr><td align="center" bgcolor="#FFFFFF"--
>

        <TABLE BORDER="0" CELLPADDING="10"
CELLSPACING="0" bgcolor="#FFFFFF" width="100%">
        <tr><td style="border-top: 1px solid
#000000; border-left: 1px solid #000000; text-align: left">
        <tiles:insert attribute='location' /><br>
        <center>
        <tiles:insert attribute='quickSearch' />
        <tiles:insert attribute='errors' />
        <tiles:insert attribute='body' />
        </center>
        <BR>&nbsp;
        </td></tr>
        </TABLE>

        <!--/td></tr>
        </table-->

        <!-- Fin contenido principal -->
        </TD>
        <!-- right -->
        <td valign="top"><tiles:insert
attribute='right' /></td>
    </tr>
    </table>

    <BR>
    <BR>&nbsp;
    </TD>
</TR>
</TABLE>
<center>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="0" WIDTH="90%"
align="center">
    <TR><TD ALIGN="CENTER" CLASS="COPY" style="border-top: #000000
1px solid;"> <tiles:insert attribute='footer' /> </TD></TR>
    </TABLE>
</center>
</body>
</html>
```

L'avantatge d'haver definit una pàgina plantilla és que podem incorporar llibreries comunes a totes les pàgines, fitxers d'estils, comportament javascript, variables, etc.

4. Annexos

4.1. Tiles

En Tiles l'element bàsic és el template. El template no és més que una pàgina JSP que usa una llibreria de tags JSP per a definir el layout d'una pàgina. El template serveix com a definició de com s'ubicaran les pàgines sense especificar el contingut. Aquest s'insereix en temps d'execució en la pàgina template. Totes les pàgines de l'aplicació poden reutilitzar el mateix template o bé usar altres.

Un exemple (*basicLayout.jsp*) de template seria:

```
<table border="0" width="100%" cellpadding="5">
<tr>
  <td colspan="2"><tiles:insert attribute="header" /></td>
</tr>
<tr>
  <td valign="top">
    <tiles:insert attribute="menu" />
  </td>
  <td valign="top" align="left">
    <table width="100%">
      <tr><td>
        <tiles:insert attribute="body" />
      </td></tr>
    </table>
  </td>
</tr>
<tr>
  <td colspan="2">
    <tiles:insert attribute="footer" />
  </td>
</tr>
</table>
```

En aquest exemple es barregen codi HTML i tags JSP. En el *tag* `<tiles:insert>` s'està indicant de manera declaratòria que el contingut a mostrar en aquella part ha de crear-se de manera dinàmica a partir del valor que s'indiqui per al valor del paràmetre 'attribute' ('header',...). El gran avantatge d'aquesta definició és que diverses pàgines de l'aplicació poden usar el mateix layout que està controlat per aquest fitxer. Si en algun moment de la vida del projecte es decideix canviar el layout només és necessari retocar aquesta pàgina única i no totes les pàgines de l'aplicació.

4.1.1. Definició d'una Pantalla basada en un Template

Una vegada definit el template en el que basarem les nostres pantalles (podem definir diversos templates amb diferents propòsits), podem definir quin és el contingut de cada pantalla.



Per a definir una pantalla es pot usar el format següent:

```
<tiles-definitions>

<definition name="nombre" path="path">
  <put name="nombreParam" value="valor" />
  ...
  <putList name="nombreList">
    <add value="valor1"/>
    <add value="valor2"/>
  </putList>
</definition>
```

On es pot fer ús dels següents atributs per a la definició:

- **name:** Nom de la pantalla. Aquest nom ha de ser únic
- **path:** Ruta a la pàgina JSP que construeix la pantalla (el template definit)

Per a especificar quins són els valors específics a substituir en el template podem usar:

- **put:** Amb cada paràmetre a passar a la definició definirem una entrada put
- **putList:** Serveix per a poder passar una paràmetre que estigui compost per una llista de valors.

Per mitjà de '`<put name="header" value="/template/header.jsp" />`' especifiquem que l'espai de la plantilla amb el codi '`<tiles:insert attribute="header" />`' serà substituït per la pàgina '/template/header.jsp'.

Definició d'una Pantalla a partir d'una altra Pantalla

És molt comú que determinades parts de les nostres pantalles siguin comunes a moltes pantalles (capçalera, peu,...). Per a cobrir aquests casos, podem especificar que una definició de pantalla estén una altra definició de pantalla.

Per a estendre des d'una pantalla una altra pantalla, s'usarà l'atribut 'extends' el valor del qual serà el nom de la definició pare.

```
<definition name="site.mainLayout" path="/layouts/basicLayout.jsp">
  <put name="title" value="Nombre Aplicacion" />
  <put name="header" value="/template/header.jsp" />
  <put name="menu" value="site.menu.bar" />
  <put name="footer" value="/template/footer.jsp" />
</definition>
...
<definition name="pages.nomPage" extends="site.mainLayout" >
  <put name="title" value="Titol" />
  <put name="body" value="/mantPagina.jsp" />
```



```
</definition>
```

En el cas mostrat, s'ha especificat una definició '*site.mainLayout*' que usa el layout definit per la pàgina '*basicLayout.jsp*'. Aquesta definició ha estat estesa per '*pages.nomPage*' especificant únicament quin serà el títol i el cos del layout, mentre que manté invariable la capçalera, menú i peu del pare.

Definir regions d'una Pantalla basades en altres Pantalles

En determinats casos, pot ser interessant que una de les parts d'una pantalla sigui una altra pantalla (basada al seu torn en una definició pare si es creu oportú).

En l'exemple anterior la definició '*site.mainLayout*' definix un paràmetre '<put name=menu>' en el que especifica que aquella regió és una altra definició de pantalla enlloc d'una pàgina JSP.

Retorn d'una Pantalla des d'Accions nostres

Per indicar el path de retorn d'una pantalla en les accions Struts usarem el nom de la definició:

```
<action      path="/..."
              type="..."
              name="..."
              scope="..."
              input="..."
              parameter="reqCode"
            >
    <forward name="success" path="pages.nomPage" />
</action>
```