

Présentation de la Taglib Displaytag

par [Eric Wawszczyk](#)

Date de publication : 14/03/2007

Dernière mise à jour : 14/03/2007

Si vous en avez assez de générer vous-même l'affichage des tableaux (TABLE en html), Displaytag est la librairie qui va vous simplifier la vie. En effet, elle permet de créer simplement et très rapidement un tableau à partir d'une liste d'objets qu'elle reçoit en paramètre. Mais ce n'est pas tout ! Elle gère aussi les tris sur les colonnes, la pagination, l'export et bien d'autres choses. Fini le travail fastidieux et répétitif !

- I - Introduction
- II - Installation
- III - Principe
- IV - Exemples d'utilisations
 - IV-A - Exemple Basic
 - IV-B - Utilisation classique
 - IV-C - Tri sur les colonnes
 - IV-D - Pagination
 - IV-E - Liens automatiques
 - IV-F - Utilisation des décorateurs
- V - Fonctions avancées
 - V-A - Configuration
 - V-B - Export des tableaux
 - V-C - Internationalisation
 - V-D - Autres fonctions
- VI - Conclusion

I - Introduction

Cet article a pour but de vous présenter la Taglib J2EE appelée Displaytag.

Pour comprendre le fonctionnement d'une Taglib, reportez-vous au cours [Présentation des JSP Tag Libraries \(Taglibs\)](#)

Dans toutes les applications web, on a systématiquement besoin d'afficher des séries d'enregistrements ou des listes d'objets via des tableaux en html (balise TABLE). Il faut alors itérer sur la liste, gérer la ligne d'entête du tableau et assurer la pagination. Cela devient vite lourd et répétitif.

En J2EE, la librairie Displaytag offre une solution rapide, pratique et efficace à ce problème. Cette Taglib offre un mécanisme très simple d'itération sur une collection, celui-ci s'exécutant directement depuis le code d'une jsp. Le résultat donne l'affichage d'un tableau html avec une quantité d'options telles que les tris sur les colonnes, la pagination, le regroupement de données ou encore l'export du tableau aux formats csv, excel, xml, pdf ou rtf.

II - Installation

En tant que Taglib, l'installation des Displaytag est très simple. La première chose à faire est bien sûr de télécharger la librairie Displaytag. Vous pourrez les trouver librement à l'url suivante : <http://displaytag.sourceforge.net>

Il suffit en fait de copier le displaytag-1.1.jar dans le répertoire `%WebAppRoot%/WEB-INF/lib` de votre application. Il faudra aussi installer les dépendances (cf. [doc d'installation](#))

Ensuite, pour pouvoir utiliser la Taglib Displaytag, il suffit de faire l'import suivant dans la page jsp :

Exemple :

```
<%@ taglib uri="http://displaytag.sf.net" prefix="display" %>
```

III - Principe

La librairie Displaytag ne fait qu'une chose : afficher des tableaux en html ! Il suffit de lui donner une liste d'objets et la librairie va gérer l'affichage des colonnes, les tris des colonnes, la pagination, les regroupements, l'export et bien d'autres choses...

La liste d'objets peut être obtenue à partir des scopes suivants : `pageScope` (page), `requestScope` (request - default), `sessionScope` (session) et `applicationScope` (application). Si votre objet liste n'est pas dans le scope par défaut "requestScope", il suffit de rajouter le nom du scope en préfixe de la liste (ex `sessionScope.mylist`)

IV - Exemples d'utilisations

Vous allez ici pouvoir comprendre comment utiliser le composant Displaytag dans votre code et quel en sera le rendu graphique.

IV-A - Exemple Basic

Voici un exemple d'utilisation très simple permettant d'afficher sous la forme d'un tableau une liste nommée "mylist" contenant des objets de type "User".

Exemple : la classe User.java

```
public class User {
    int id;
    String nom;
    String prenom;
    String email;
    int age;

    // getters and setters
}
```

Exemple : jsp

```
<display:table name="mylist" />
```

Et le résultat :

Age	Prenom	Email	Nom	Id
34	Dupond	edupond@nowhere.fr	Eric	1
38	Durand	jdurand@nowhere.fr	Jean	2
55	Dupond	ldupond@nowhere.fr	Lucien	3
6	Auchon	edupond@nowhere.fr	Paul	4
3	Breton	breton@nowhere.fr	Clément	5
15	Pagis	apagis@nowhere.fr	Alphonse	6

IV-B - Utilisation classique

Cet exemple permet de préciser les colonnes que vous souhaitez afficher et les noms des entêtes de ces colonnes.

Il suffit de rajouter le tag `<display:column property="...">`. property étant le nom de l'attribut dans le bean User.

Exemple : jsp

```
<display:table name="mylist" >
  <display:column property="id" title="N°" />
  <display:column property="nom" title="Nom" />
  <display:column property="prenom" title="Prénom" />
  <display:column property="email" title="Email" />
</display:table>
```

Et le résultat :

N°	Nom	Prénom	Email
1	Eric	Dupond	edupond@nowhere.fr
2	Jean	Durand	jdurand@nowhere.fr
3	Lucien	Dupond	ldupond@nowhere.fr
4	Paul	Auchon	edupond@nowhere.fr
5	Clément	Breton	breton@nowhere.fr
6	Alphonse	Pagis	apagis@nowhere.fr

IV-C - Tri sur les colonnes

Il est possible de créer des colonnes triables. Le composant Displaytag assure un tri "en mémoire" de la liste. Cela est très donc très simple d'utilisation et très rapide pour de petites listes. Cela ne convient donc pas aux (très) grandes listes. Dans ce cas, il faudra plutôt paramétrer le composant pour une "pagination externe" avec un tri effectué par la base de données par exemple.

Pour rendre une colonne triable, il suffit d'ajouter **sortable="true"**.

Pour préciser le tri par défaut du tableau, il faut ajouter **defaultsort="..."**. defaultsort étant le numéro de la colonne à trier.

Exemple : jsp

```
<display:table name="mylist" defaultsort="2" defaultorder="descending" >
  <display:column property="id" title="N°" />
  <display:column property="nom" title="Nom" sortable="true" />
  <display:column property="prenom" title="Prénom" />
  <display:column property="email" title="Email" sortable="true" />
</display:table>
```

Et le résultat :

N°	Nom	Prénom	Email
4	Paul	Auchon	edupond@nowhere.fr
3	Lucien	Dupond	ldupond@nowhere.fr
2	Jean	Durand	jdurand@nowhere.fr
1	Eric	Dupond	edupond@nowhere.fr
5	Clément	Breton	breton@nowhere.fr
6	Alphonse	Pagis	apagis@nowhere.fr

Si vous utilisez Struts (ou une servlet) pour initialiser votre jsp, il vous faudra alors rajouter l'attribut **requestURI="/monAction.do"** pour que la page soit repostée sur l'action et non sur la jsp. Quand cet attribut est présent, les liens auto générés pour les tris, les exports et la pagination sont basés sur ce paramètre.

IV-D - Pagination

La gestion de la pagination est entièrement prise en charge par le composant Displaytag. Il suffit de préciser le

nombre d'éléments à afficher par page. Le composant va alors générer automatiquement la numérotation et la pagination (liens page précédente, page suivante, première page et dernière page).

Il suffit d'ajouter **pagesize="..."** pour préciser le nombre de lignes à afficher par page

Exemple : jsp

```
<display:table name="mylist" pagesize="5" />
```

Et le résultat :

6 items found, displaying 1 to 5.
[First/Prev] 1, 2 [Next/Last]

N°	Nom	Prénom	Email
4	Paul	Auchon	edupond@nowhere.fr
3	Lucien	Dupond	ldupond@nowhere.fr
2	Jean	Durand	jdurand@nowhere.fr
1	Eric	Dupond	edupond@nowhere.fr
5	Clément	Breton	breton@nowhere.fr

En rajoutant les attributs **offset="..." length="..."** par exemple, vous pourrez même restreindre l'affichage à une partie de la liste (ex : offset="3" length="5")

IV-E - Liens automatiques

Si vous avez des adresses Emails ou des Urls Web dans une colonne et que vous souhaitez afficher ces données comme des liens hypertextes, il suffit de rajouter l'attribut **autolink="true"**. Displaytag va générer pour vous ces liens.

Exemple : jsp

```
<display:table name="mylist" >
  <display:column property="id" title="N°" />
  <display:column property="nom" title="Nom" />
  <display:column property="prenom" title="Prénom" />
  <display:column property="email" title="Email" autolink="true" />
</display:table>
```

Et le résultat :

N°	Nom	Prénom	Email
4	Paul	Auchon	edupond@nowhere.fr
3	Lucien	Dupond	ldupond@nowhere.fr
2	Jean	Durand	jdurand@nowhere.fr
1	Eric	Dupond	edupond@nowhere.fr
5	Clément	Breton	breton@nowhere.fr

Le tag column fournit également 5 "struts-like" attributs que vous pouvez utiliser pour créer un lien dynamique (href, paramID, paramName, paramProperty, paramScope).

Exemple : jsp

```
<display:table name="mylist" >
  <display:column property="id" title="N°" />
  <display:column property="nom" title="Nom" href="details.jsp" paramId="idUtil"
paramProperty="id" />
  <display:column property="prenom" title="Prénom" />
  <display:column property="email" title="Email" />
</display:table>
```

IV-F - Utilisation des décorateurs

La librairie offre un autre mécanisme appelé Decorator. Il s'agit en fait de classe héritant de la classe TableDecorator.

Dans ces classes on définit des méthodes qui seront appelées lors de la génération des tables.

Un mapping sera automatiquement effectué entre les attributs property des tags column et ces méthodes.

On peut ainsi facilement modifier le rendu de l'affichage d'une colonne. Par exemple, lorsque l'on veut afficher des monnaies ou des dates ou encore si l'on veut générer dynamiquement des liens html.

Exemple : la classe MyDecorator.java

```
public class MyDecorator extends TableDecorator {

    public String getMyLink()
    {
        User user= (User) getCurrentRowObject();
        int lId= user.getId();

        return "<a href=\"details.jsp?idUtil=" + lId + "\">Modifier</a>";
    }
}
```

Exemple : jsp

```
<display:table name="mylist" decorator="MyDecorator" />
  <display:column property="id" title="N°" />
  <display:column property="nom" title="Nom" />
  <display:column property="prenom" title="Prénom" />
  <display:column property="email" title="Email" />
  <display:column property="myLink" title="" />
</display:table>
```

Et le résultat :

N°	Nom	Prénom	Email	
4	Paul	Auchon	edupond@nowhere.fr	Modifier
3	Lucien	Dupond	ldupond@nowhere.fr	Modifier
2	Jean	Durand	jdurand@nowhere.fr	Modifier
1	Eric	Dupond	edupond@nowhere.fr	Modifier
5	Clément	Breton	breton@nowhere.fr	Modifier
6	Alphonse	Pagis	apagis@nowhere.fr	Modifier

V - Fonctions avancées

V-A - Configuration

La configuration par défaut de Displaytag est définie dans le fichier `displaytag.properties` contenu dans la librairie `displaytag.jar`.

Il y a 2 possibilités pour modifier cette configuration :

- pour toute l'application web, créer un fichier personnalisé nommé **displaytag.properties** et le sauvegarder dans le classpath (`WEB-INF/classes`)
- pour un tableau spécifique, en utilisant dans le tableau les tags **<display:setProperty ...>**

Exemple : `displaytag.properties`

```
basic.empty.showtable=true
basic.msg.empty_list=No results matched your criteria.

#paging.banner.placement=top
paging.banner.placement=bottom

export.types=csv excel xml pdf rtf
export.excel=true
export.csv=true
export.xml=true
export.pdf=true
export.rtf=true
export.excel.class=org.displaytag.export.ExcelView
export.pdf.class=org.displaytag.export.DefaultPdfExportView
export.rtf.class=org.displaytag.export.DefaultRtfExportView
# if set, file is downloaded instead of opened in the browser window
#export.[mymedia].filename=
```

Pour avoir la liste de toutes les options configurables, reportez-vous à la [documentation](#)

V-B - Export des tableaux

Pour exporter les tableaux, il faut copier dans le répertoire `%WebAppRoot%/WEB-INF/lib` de votre application, les librairies `itext.jar` et `displaytag-export-poi.jar`.

Quand vous précisez le flag **export="true"** dans le tag `table`, le composant offre alors la possibilité d'exporter le tableau affiché (WYSIWYG) sous la forme d'un fichier au format csv, excel, xml, pdf ou rtf.

En spécifiant le tag **media="..."**, vous pouvez préciser quelles colonnes seront exportées et dans quelle sortie.

Par exemple `media="html excel"` permet de n'afficher la colonne que dans les pages html et dans les fichiers excel

Exemple : jsp

```
<display:table name="mylist" export="true" />
```

Et le résultat :

N°	Nom	Prénom	Email	
4	Paul	Auchon	edupond@nowhere.fr	Modifier
3	Lucien	Dupond	ldupond@nowhere.fr	Modifier
2	Jean	Durand	jdurand@nowhere.fr	Modifier
1	Eric	Dupond	edupond@nowhere.fr	Modifier
5	Clément	Breton	breton@nowhere.fr	Modifier
6	Alphonse	Pagis	apaqis@nowhere.fr	Modifier

Export options: CSV | Excel | XML | PDF | RTF

Lors de l'export, si vous rencontrez des erreurs liées au content type, il faudra installer un filtre. Reportez-vous à la [documentation](#)

V-C - Internationalisation

Si vous souhaitez utiliser Displaytag dans une application multi-langues, vous aurez besoin de traduire les messages et libellés html utilisés par le composant. Pour cela, il suffit de créer un fichier par langue et de les nommer `displaytag_LANGUAGE.properties` (ex : `displaytag_en.properties`, `displaytag_it.properties`).

Si vous utilisez Displaytag avec Struts, vous pourrez utiliser les fichiers `messages.properties` de Struts pour gérer les traductions des entêtes de colonnes. Pour cela ajouter la ligne suivante dans le fichier de configuration de Displaytag :

Exemple : `displaytag.properties`

```
locale.provider=org.displaytag.localization.I18nStrutsAdapter
```

Le tag **titleKey** permet pour chaque colonne de préciser le code du libellé correspondant dans le fichier `messages.properties` ou `displaytag.properties` :

Exemple : `message.properties`

```
utilisateurs.text.TitreColId=N°
utilisateurs.text.TitreColNom=Nom
utilisateurs.text.TitreColPrenom=Prénom
utilisateurs.text.TitreColPrenom=Email
```

Exemple : `jsp`

```
<display:table name="mylist" >
  <display:column property="id" titleKey="utilisateurs.text.TitreColId" />
  <display:column property="nom" titleKey="utilisateurs.text.TitreColNom" />
  <display:column property="prenom" titleKey="utilisateurs.text.TitreColPrenom" />
  <display:column property="email" titleKey="utilisateurs.text.TitreColEmail" />
</display:table>
```

V-D - Autres fonctions

Les Displaytag offrent de nombreuses autres possibilités :

- lorsque vous avez une liste dont les objets sont triés et regroupés sur plusieurs colonnes : au lieu de répéter les mêmes valeurs dans toutes ces colonnes, seules les valeurs qui changent seront affichées,

- la totalisation d'une colonne,
- un entête (header) et un bas de page (footer) au tableau,
- le formatage des dates et des nombres,
- la pagination externe. Cela permet de gérer manuellement par programmation le tri et la pagination. [Lire cet article.](#)

VI - Conclusion

Les principales fonctionnalités apportées par cette librairie sont impressionnantes :

- rapidité et facilité d'utilisation,
- tri sur les colonnes,
- gestion de la pagination,
- export aux formats csv, excel, xml, pdf ou rtf.

Couplée à Struts, cette librairie nous offre aujourd'hui un gain majeur en rapidité de développement et en maintenance.

Merci à ma société [Webnet](#) pour m'avoir laissé du temps pour rédiger cet article.

Télécharger :

[Télécharger les sources et le war du tutoriel \(Miroir\)](#)

Liens :

Le site Displaytag : <http://displaytag.sourceforge.net>

Les exemples : <http://displaytag.homeip.net/displaytag-examples-1.1/>