

DisplayTag

A DisplayTag é uma suite open source de tags customizadas que fornecem padrões de apresentação de alto nível. A sua utilização é simples pelo fato dela utilizar o modelo MVC.

Funcionalidades

A DisplayTag provê uma série de funcionalidades. Algumas delas são:

- Exportação para Excel, CSV e XML.
- Ordenação.
- Paginação.
- Utilização de CSS.
- Decorators.

Instalação

O primeiro passo é baixar o JAR da DisplayTag no seguinte endereço:

<http://prdownloads.sourceforge.net/displaytag/displaytag-1.1-bin.zip?download>

Após baixar o JAR, é necessário baixar as dependências:

- [commons-logging](#)
- [commons-lang](#)
- [commons-collections](#)
- [commons-beanutils](#)
- [log4j](#)
- [itext](#) (opcional, para exportação pdf/rtf)

Basta adicionar os JARs ao classpath e pronto !

Hello World com DisplayTag

Utilizaremos o WebWork como controlador e o Hibernate para persistência. Detalhes do funcionamento dos frameworks citados está fora do escopo deste tutorial.

Para o nosso Hello World, precisamos criar uma simples Action que realize uma pesquisa na base de dados e retorne todos os alunos cadastrados:

Por Guilherme Mello do Nascimento (JavaPlayer)

```
package com.portaljava.action;
import java.util.ArrayList;
import java.util.List;

import com.opensymphony.xwork.ActionSupport;
import com.portaljava.dao.GenericDAO;
import com.portaljava.model.Aluno;
import com.portaljava.util.DAOFactory;

public class ListAction extends ActionSupport {
    private static final String SUCCESS = "SUCCESS";
    private GenericDAO dao = DAOFactory.getDAO("aluno");
    private List list = new ArrayList();

    public String execute() {
        list = dao.findAll(Aluno.class);
        return SUCCESS;
    }

    // get e sets para list
}
```

E no JSP, o código para exibir a DisplayTag fica assim:

```
<display:table name="${list}">
    <display:column property="idAluno" title="ID"/>
    <display:column property="nmAluno" title="Nome"/>
</display:table>
```

A seguir, uma descrição de cada atributo utilizado no exemplo acima:

<display:table> - Gera uma tabela HTML.

name – Faz referência ao objeto que irá popular a tabela.

<display:column> - Exibe uma propriedade de um objeto em uma linha dentro da tabela.

property – Armazena o valor de uma propriedade de um Objeto.

title – Exibe um título para a coluna.

Ordenação

Uma das funcionalidades mais interessantes da DisplayTag é a possibilidade de ordenar por colunas. Para ordenar por uma coluna basta adicionar os seguintes atributos:

<display:table>

sort – Indica o tipo de ordenação. Aceita os seguintes parâmetros:

list – Toda a lista é ordenada.

page – apenas a página corrente é ordenada.

Por Guilherme Mello do Nascimento (JavaPlayer)

<display:column>

sortable – Indica se a coluna aceita ordenação. Aceita os seguintes parâmetros:
true – A coluna aceita ordenação.
false – A coluna não aceita ordenação.

Caso o usuário não selecione uma coluna, podemos deixar uma coluna ordenada por default. Para isso, devemos indicar a coluna a ser ordenada na tag <display:table>

<display:table>

defaultsort – Indica a coluna que receberá ordenação por default. Os índices das colunas começam em zero.

Ao adicionar suporte a ordenação, a listagem deve ficar assim:

```
<display:table name="list" sort="list" defaultsort="1">  
  <display:column property="idAluno" title="ID" sortable="true"/>  
  <display:column property="nmAluno" title="Nome" sortable="true"/>  
</display:table>
```

Paginação

A DisplayTag oferece um ótimo suporte a paginação de dados. Para que sua lista tenha acesso a essa funcionalidade, adicione os seguintes atributos:

<display:table>

pagesize – Indica o número de linhas por página. Você deve inserir o número de linhas que você deseja exibir por página.

requestURI – Indica a URL em que a paginação será realizada.

Agora já podemos adicionar o suporte a paginação em nossa listagem:

```
<display:table name="list" sort="list" defaultsort="1" pagesize="5"  
requestURI="display_list.action">  
  
  <display:column property="idAluno" title="ID" sortable="true"/>  
  <display:column property="nmAluno" title="Nome" sortable="true"/>  
  
</display:table>
```

Exportação

Outra funcionalidade que a DisplayTag fornece é a de exportação de dados. Os dados podem ser exportados para o seguinte formatos:

- CVS
- Excel
- XML
- PDF

Para que sua listagem tenha acesso ao suporte de exportação, você deve adicionar os seguintes atributos:

<display:table>

- export – Indica se a listagem poderá exportar os dados.
- true - A listagem poderá exportar.
- false – A listagem não poderá exportar.

A listagem ficará assim:

```
<display:table name="list" sort="list" defaultsort="1" pagesize="5"
requestURI="display_list.action" export="true">

    <display:column property="idAluno" title="ID" sortable="true"/>
    <display:column property="nmAluno" title="Nome" sortable="true"/>

</display:table>
```

Decorando sua Listagem

A DisplayTag utiliza o Design Pattern Decorator. Um Decorator é usado para adicionar funcionalidades em um objeto particular. O Decorator também é conhecido como Wrapper, que por sua vez é um objeto que possui uma interface idêntica ao objeto que ele contém.

Para utilizar um decorator, precisamos criar um atributo na objeto que populará a displaytag:

```
public class Aluno {

    private long idAluno;
    private String nmAluno;
    private int idade;

    // atributo que será decorado
    private String operacoes;

    public Aluno() {
    }

    public Long getIdAluno() {
        return idAluno;
    }
    public void setIdAluno(Long idAluno) {
        this.idAluno = idAluno;
    }
    public String getNmAluno() {
        return nmAluno;
    }
}
```

Por Guilherme Mello do Nascimento (JavaPlayer)

```
public void setIdade(int idade) {
    this.idade = idade;
}
public String getOperacoes() {
    return operacoes;
}
public void setOperacoes(String operacoes) {
    this.operacoes = operacoes;
}
}
```

Agora devemos criar uma classe que estenda a classe TableDecorator.

Para que a displaytag seja decorada, devemos criar no decorator um método get para o atributo que foi criado no Bean. Ou seja, como criamos no Bean o atributo *operacoes*, devemos criar um método *getOperacoes()*.

```
public class Wrapper extends TableDecorator {

    public String getOperacoes() throws Exception {
        // pega o objeto que está na linha corrente
        Object objeto = this.getCurrentRowObject();
        String idAluno;

        if (objeto instanceof Aluno) {
            Aluno aluno = (Aluno) objeto;

            // pega o id do aluno
            idAluno = aluno.getIdAluno().toString();
        } else {
            throw new Exception("Objeto não encontrado.");
        }
        /* retorna uma String que adicionará um link na displaytag.
        Você poderá decorar no jeito que quiser, adicionando um link (a href), um textarea,
        textfield, checkbox e etc.
        */
        return "<a href=\"delete.action?aluno.idAluno=\"" + idAluno + "\">" + "Excluir</a>";
    }
}
```

Para utilizar o decorator na displaytag basta utilizar o atributo *decorator*, passando o pacote completo da classe.

Por Guilherme Mello do Nascimento (JavaPlayer)

```
<display:table name="list" decorator="com.portaljava.decorator.Wrapper " sort="list"
defaultsort="0" requestURI="display_list.action" pagesize="5" export="true">

  <display:column property="idAluno" title="ID"/>
  <display:column property="idade" title="Idade" sortable="true"
    style="text-align:center;"/>
  <display:column property="nmAluno" title="Nome" sortable="true" style="width:300px;"/>
  <display:column property="operacoes" title="Operações"/>

</display:table>
```

No atributo `decorator` informamos o pacote onde o classe decorator encontra-se e criamos mais uma coluna para receber a nova funcionalidade, apontando para o atributo `operacoes`.

Personalizando

Podemos personalizar a nossa displaytag de diversas maneiras. Basta criar um arquivo `properties` chamado `displaytag.properties`. É nesse arquivo que se encontra as configurações do que irá aparecer na tela. Essas informações se encontram por default em inglês.

O arquivo `properties` deve ficar armazenado em `WEB-INF/src` para que possa sobrepor o arquivo `properties` que está no JAR da displaytag.

No `properties` podemos configurar a posição da paginação, mensagens, CSS, ordenação, etc. Para maiores informações:

<http://displaytag.sourceforge.net/10/configuration.html>

Utilizando CSS

A display tag possui o suporte a CSS. A tag **<display:table>** cria tabelas HTML com seções: `<thead>` e `<tbody>`. As classes CSS são adicionadas automaticamente quando necessário.

A seguir, uma lista com as classes CSS que podem ser aplicadas a sua displaytag:

<i>classe</i>	<i>funcionalidade</i>
odd e even	Classes para atribuir um cor em cada linha de dados.
sorted	Classe atribuída a tag <i>tr</i> de uma coluna ordenada.
order1	Classe atribuída a tag <i>th</i> de uma coluna ordenada se a ordem for ascendente.
order2	Classe atribuída a tag <i>th</i> de uma coluna ordenada se a ordem for descendente.
sortable	Classe atribuída a tag <i>th</i> de uma coluna ordenável.

Por Guilherme Mello do Nascimento (JavaPlayer)

Até a próxima!

Comentários, críticas ou sugestões: javaplayer@gmail.com
Qualquer dúvida, poste no [fórum](#) do Portal Java.

Sobre o autor: Guilherme Mello do Nascimento é moderador e freqüentador assíduo do Portal Java. Trabalha com Java para Web a dois anos.

Referências:

<http://displaytag.sourceforge.net/11/>