Displaying Results Sets with Data Grids

Table of contents

Copyright © 2004 The Apache Software Foundation. All rights reserved.

1. Database-Driven Web Applications

A database-driven Beehive web application has the the follow components:

- one or more page flows (= a controller file + JSP pages)
- one or more database control files
- one or more databases

A basic schema for a database-driven application is shown below.

pageflow_database

The Database

The role of the database is as a storehouse for the data. The database persists the data from session to session and is the common object which users view and operate upon.

The Database Control

The database control handles the data traffic with the database. The database control is a Java class that handles the database connection and handles the individual operations on the database. Typically, the methods in the class have associated SQL statements: when a method is called, the associated SQL statement is sent to the database. Any data returned by the database is transformed by the method into an appropriate Java object so it is available to the rest of the web application.

For more database control implementations you can download and use in applications, see Database Control Sample (../controls/sample_controls-db.html) and Controlhaus's JDBC Control (http://jdbc.controlhaus.org/)

The Page Flow

The Page Flow(s) form the front-end user interface of the application. Through the JSP pages, users can add, delete, update, and view the data in the database. Beehive provides a specialized tag libarary for viewing and managing the complex data sets associated with database applications: `<netui-data:xxx>` (../apidocs/taglib/taglib-overview-summary.html#netui-data) .

The remainder of this topic concentrates on using the <netui-data:xxx> tab library to handle data sets.

2. Setting Up a Basic Grid

This section takes you through a step by step process of setting up a basic grid for 'tabular' data, data such as a ResultSet, RowSet, or any kind of data that can easily be set out like table

Copyright © 2004 The Apache Software Foundation. All rights reserved.

with rows and columns.

2.1. Setup the Pestore Sample

To set up a grid you first need a data source of some tabular data. Below we use the Petstore sample as the source of tabluar data. Begin by running through the set up instructions for the Petstore (../jpetstore.html) Sample.

2.2. Connect to a Tabular Data Source

Now that the Petstore sample is up and running, we are ready to use its data sources.

Edit the Controller class

Edit the `viewCategory()` method of the file `petstoreWeb/shop/Controller.jpf` as follows. Code to edit is shown in bold.

`petstoreWeb/shop/Controller.jpf`

```
    @Jpf.Action(
        forwards = {
            @Jpf.Forward(name = "category", path = "categoryGrid.jsp",
                         actionOutputs = {
                             @Jpf.ActionOutput(name = "category",
                                               type =
org.apache.beehive.samples.petstore.model.Category.class,
                                               required = true),
                             @Jpf.ActionOutput(name = "products",
                                               type =
org.apache.beehive.samples.petstore.model.Product[].class,
                                               required = false)
                         })
        }
    )
    protected Forward viewCategory() {
    ...
    }
```

Add a JSP page

Add a page called `categoryGrid.jsp` to the folder `petstoreWeb\shop\`. Edit the file so it appears as follows.

`petstoreWeb/shop/categoryGrid.jsp`

```
<%@ page language="java" contentType="text/html;charset=UTF-8"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-databinding-1.0"
prefix="netui-data"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-html-1.0"
```


Copyright © 2004 The Apache Software Foundation. All rights reserved.

```
prefix="netui"%>

<netui-data:declarePageInput name="products"
type="org.apache.beehive.samples.petstore.model.Product[]"
required="false"/>
```

You now have an action method (`viewCategory()`) that sends tabular data to a JSP page (`categoryGrid.jsp`). The tabular data is available through the databinding context `pageInput.products`. In the next step you will point a data grid at this databinding context.

Recompile Petstore

Before proceeding, recompile Petstore.

The recompile step is necessary because you have modified the Controller.jpf file. In the steps below, where you only modify the JSP page `categoryGrid.jsp`, you will not need to recompile the Petstore app to view your changes.

2.3. Render a Basic Data Grid

Edit `categoryGrid.jsp` so it appears as follows.

`petstoreWeb/shop/Controller.jpf`

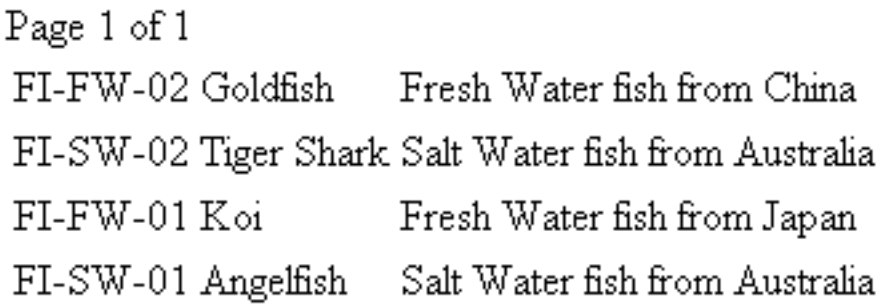
```
<%@ page language="java" contentType="text/html;charset=UTF-8"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-databinding-1.0"
prefix="netui-data"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-html-1.0"
prefix="netui"%>

<netui-data:declarePageInput name="products"
type="org.apache.beehive.samples.petstore.model.Product[]"
required="false"/>

<netui-data:dataGrid name="productsGrid" dataSource="pageInput.products">
    <netui-data:rows>
        <netui-data:spanCell value="${container.item.productId}" />
        <netui-data:spanCell value="${container.item.name}" />
        <netui-data:spanCell value="${container.item.description}"/>
    </netui-data:rows>
</netui-data:dataGrid>
```

When you visit the URL

```
http://localhost:8080/petstoreWeb/shop/viewCategory.do?catId=FISH
```

you should see the following page

Copyright © 2004 The Apache Software Foundation. All rights reserved.



categoryGrid1

The tabular data is rendered as an HTML table, by the remaining tags (<netui-data:rows> and <netui-data:spanCell).

Here is the renderd HTML table.

```
Page 1 of 1
<table class="datagrid">
    <tr class="datagrid-even">
        <td class="datagrid"><span>FI-FW-02</span></td>
        <td class="datagrid"><span>Goldfish</span></td>
        <td class="datagrid"><span>Fresh Water fish from China</span></td>
    </tr>
    <tr class="datagrid-odd">
        <td class="datagrid"><span>FI-SW-02</span></td>
        <td class="datagrid"><span>Tiger Shark</span></td>
        <td class="datagrid"><span>Salt Water fish from
Australia</span></td>
    </tr>
    <tr class="datagrid-even">
        <td class="datagrid"><span>FI-FW-01</span></td>
        <td class="datagrid"><span>Koi</span></td>
        <td class="datagrid"><span>Fresh Water fish from Japan</span></td>
    </tr>
    <tr class="datagrid-odd">
        <td class="datagrid"><span>FI-SW-01</span></td>
        <td class="datagrid"><span>Angelfish</span></td>
        <td class="datagrid"><span>Salt Water fish from
Australia</span></td>
    </tr>
</table>
```

Notice that the tabular data is passed to the data grid through the `dataSource` attribute:

```
<netui-data:dataGrid name="productsGrid" dataSource="pageInput.products">
```

When a data set is passed into a <netui-data:dataGrid> tag, that data becomes accessible through the databinding contexts **`container`**, **`item`**, and **`index`**.

- **`container`** refers to the data set passed into the data grid.

Copyright © 2004 The Apache Software Foundation. All rights reserved.

- **`container.item`** refers to the current row of the data set. `container.item` refers to each row as the data grid iterates through the set.
- **`container.item.[field_name]`** refers to an individual field of the current row.
- **`container.index`** refers to the integer index of the current row.

The <netui-data:rows> tag controls the rendering of the <tr> tags. Through the <netui-data:rows> tag, you can control the attributes of the rendered <tr> tags, attribute such as `align`, `valign`, and `style`.

In a similar way, the <netui-data:spanCell> tag controls the rendering of the <td> tags.

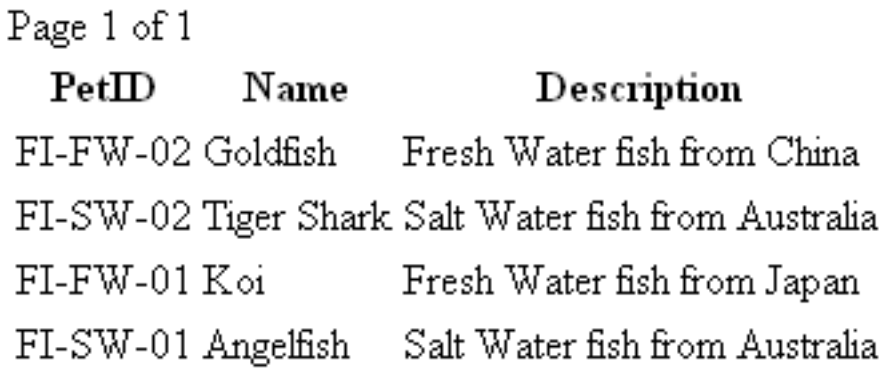
2.4. Adding a Header

Edit `categoryGrid.jsp` so it appears as follows.

```
<%@ page language="java" contentType="text/html;charset=UTF-8"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-databinding-1.0"
prefix="netui-data"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-html-1.0"
prefix="netui"%>

<netui-data:declarePageInput name="products"
type="org.apache.beehive.samples.petstore.model.Product[]"
required="false"/>

<netui-data:dataGrid name="productsGrid" dataSource="pageInput.products">
    <netui-data:header>
        <netui-data:headerCell headerText="PetID"/>
        <netui-data:headerCell headerText="Name"/>
        <netui-data:headerCell headerText="Description"/>
    </netui-data:header>
    <netui-data:rows>
        <netui-data:spanCell value="${container.item.productId}" />
        <netui-data:spanCell value="${container.item.name}" />
        <netui-data:spanCell value="${container.item.description}"/>
    </netui-data:rows>
</netui-data:dataGrid>
```

Refresh the browser. You should see the following page.

Copyright © 2004 The Apache Software Foundation. All rights reserved.



categoryGrid2

The <netui-data:headerCell> renders the following HTML <tr><th> row.

```
<tr class="datagrid-header">
    <th class="datagrid">PetID</th>
    <th class="datagrid">Name</th>
    <th class="datagrid">Description</th>
</tr>
```

2.5. Controlling the CSS Style

Notice the various `class` attributes in the rendered HTML.

- `class="datagrid"`
- `class="datagrid-header"`
- `class="datagrid-even"`
- `class="datagrid-odd"`

You control the data grid's style by changing the *prefix* rendered in the `class` attributes. If no prefix value is explicitly specified, the default prefix value is *datagrid*.

To put this into practice, first create the following CSS file.

jpetstoreWeb/resource/css/style.css

```
/* Table row style referenced in generated various generated pages */
.gridStyle
{
color: #111111;
font-size: 14px;
text-align: left;
vertical-align: middle;
padding: 5px 5px 5px 5px;
}
.gridStyle-header
{
```


Copyright © 2004 The Apache Software Foundation. All rights reserved.

```
background-color: #aaddaa;
color: #FFF;
vertical-align: baseline;
line-height: 18px;
}
.gridStyle-even
{
background-color: #FFFFFF;
color: #111111;
font-size: 14px;
text-align: left;
border-color: #999999;
border-style: solid;
border-width: 1px;
padding-left: 12px;
}
.gridStyle-odd
{
background-color: #dddddd;
color: #111111;
font-size: 14px;
text-align: left;
border-color: #999999;
border-style: solid;
border-width: 1px;
padding-left: 12px;
}
```

To reference this CSS file, edit `categoryGrid.jsp` as shown below.

petstoreWeb/shop/categoryGrid.jsp

```
<%@ page language="java" contentType="text/html;charset=UTF-8"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-databinding-1.0"
prefix="netui-data"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-html-1.0"
prefix="netui"%>

<link href="<%=request.getContextPath()%>/resources/css/style.css"
type="text/css" rel="stylesheet"/>

<netui-data:declarePageInput name="products"
type="org.apache.beehive.samples.petstore.model.Product[]"
required="false"/>

<netui-data:dataGrid name="productsGrid" dataSource="pageInput.products"
styleClassPrefix="gridStyle">
    <netui-data:header>
        <netui-data:headerCell headerText="PetID"/>
        <netui-data:headerCell headerText="Name"/>
        <netui-data:headerCell headerText="Description"/>
    </netui-data:header>
    <netui-data:rows>
        <netui-data:spanCell value="${container.item.productId}" />
        <netui-data:spanCell value="${container.item.name}" />
```


Copyright © 2004 The Apache Software Foundation. All rights reserved.

```
        <netui-data:spanCell value="${container.item.description}"/>
    </netui-data:rows>
</netui-data:dataGrid>
```

The resulting HTML page is shown below.

Page 1 of 1

PetID	Name	Description
FI-FW-02	Goldfish	Fresh Water fish from China
FI-SW-02	Tiger Shark	Salt Water fish from Australia
FI-FW-01	Koi	Fresh Water fish from Japan
FI-SW-01	Angelfish	Salt Water fish from Australia

categoryGrid3

2.6. Configuring the Pager

The "pager" controls how many rows are included on a single page of data.

The pager can be configured to appear above and/or below the data grid.

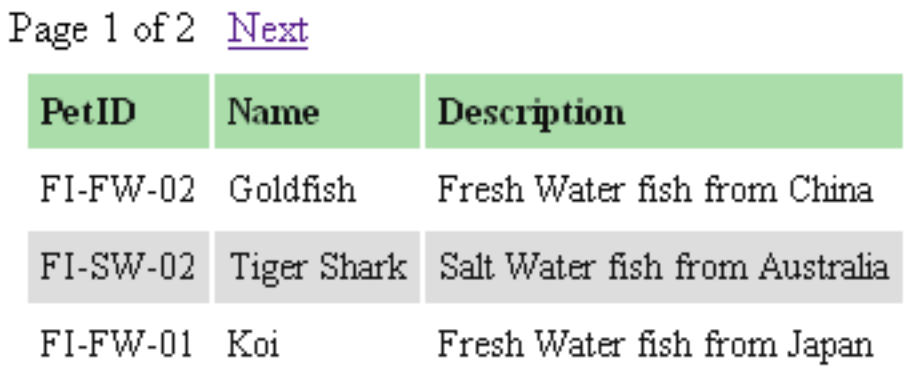
To add a pager, edit `categoryGrid.jsp` so it appears as follows.

```
<%@ page language="java" contentType="text/html;charset=UTF-8"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-databinding-1.0"
prefix="netui-data"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-html-1.0"
prefix="netui"%>
<link href="<%=request.getContextPath()%>/resources/css/style.css"
type="text/css" rel="stylesheet"/>

<netui-data:declarePageInput name="products"
type="org.apache.beehive.samples.petstore.model.Product[]"
required="false"/>

<netui-data:dataGrid name="productsGrid" dataSource="pageInput.products"
styleClassPrefix="gridStyle">
    <netui-data:configurePager pageSize="3" pagerFormat="prevNext"
pageAction="viewCategory.do"/>
    <netui-data:header>
        <netui-data:headerCell headerText="PetID"/>
```


Copyright © 2004 The Apache Software Foundation. All rights reserved.

```
        <netui-data:headerCell headerText="Name"/>
        <netui-data:headerCell headerText="Description"/>
    </netui-data:header>
    <netui-data:rows>
        <netui-data:spanCell value="${container.item.productId}" />
        <netui-data:spanCell value="${container.item.name}" />
        <netui-data:spanCell value="${container.item.description}"/>
    </netui-data:rows>
</netui-data:dataGrid>
```

Refresh the browser. You will see the following page.



categoryGrid4

2.7. Adding Linking Columns

Linking columns allow you to link to details pages for individual items in the data set.

The <netui-data:anchorCell tag adds a linking column to the data grid.

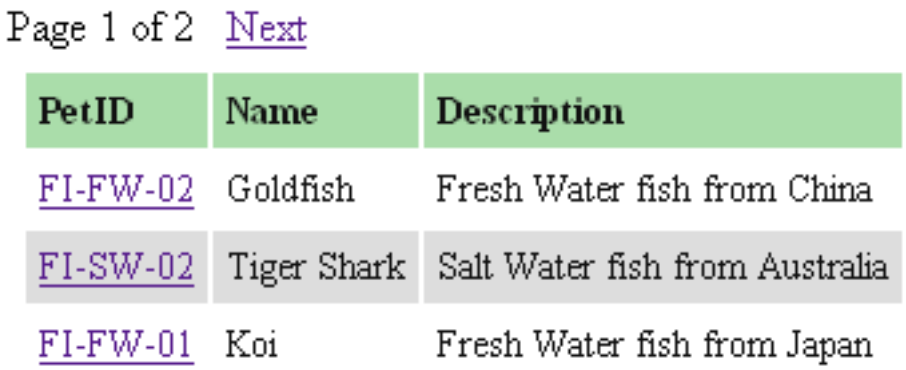
The child <parameter> tag adds the string `productId=[some_value]` to the URL.

```
<%@ page language="java" contentType="text/html;charset=UTF-8"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-databinding-1.0"
prefix="netui-data"%>
<%@ taglib uri="http://beehive.apache.org/netui/tags-html-1.0"
prefix="netui"%>
<link href="<%=request.getContextPath()%>/resources/css/style.css"
type="text/css" rel="stylesheet"/>

<netui-data:declarePageInput name="products"
type="org.apache.beehive.samples.petstore.model.Product[]"
required="false"/>

<netui-data:dataGrid name="productsGrid" dataSource="pageInput.products"
styleClassPrefix="gridStyle">
    <netui-data:configurePager pageSize="3" pagerFormat="prevNext"
pageAction="viewCategory.do"/>
```


Copyright © 2004 The Apache Software Foundation. All rights reserved.

```
    <netui-data:header>
        <netui-data:headerCell headerText="PetID"/>
        <netui-data:headerCell headerText="Name"/>
        <netui-data:headerCell headerText="Description"/>
    </netui-data:header>
    <netui-data:rows>
        <netui-data:anchorCell action="viewProduct"
value="${container.item.productId}">
            <netui:parameter name="productId"
value="${container.item.productId}"/>
        </netui-data:anchorCell>
        <netui-data:spanCell value="${container.item.name}" />
        <netui-data:spanCell value="${container.item.description}"/>
    </netui-data:rows>
</netui-data:dataGrid>
```

Notice that the PetId column now appears as linking text.



categoryGrid5

2.8. Filtering a Data Grid

[todo]

2.9. Sorting a Data Grid

[todo]

Copyright © 2004 The Apache Software Foundation. All rights reserved.