

Repeating JSP Tags

Table of contents

| | |
|----------------------------|---|
| 1 Overview..... | 2 |
| 2 Repeater..... | 2 |
| 2.1 Read only Data..... | 2 |
| 2.2 Read / write Data..... | 3 |
| 3 Cell Repeater..... | 4 |

1. Overview

The NetUI JSP tag library contains a set of JSP tags used to repeat over a data set rendering HTML markup for each item in the set. There are three sets of these tags:

- Repeater -- used to render table rows, lists, and other repeating blocks of markup
- Cell Repeater -- used to render markup in cells of a table of fixed or variable size
- Data Grid -- used to render complex pageable, sortable, filterable data grids

2. Repeater

The repeater is used similar to a `for` loop for iterating over a data set. In general, the repeater renders blocks of markup repeatedly for each item in a data set. It is also possible to create a header and footer inside of the repeater that render before and after the data set is rendered, respectively. Each of these regions is offset from the others by JSP tags that mark *rendering boundaries* inside of the repeater. Repeaters are often used to render table rows, ordered and unordered lists, repeating `divs` and so on. The [cell repeater](#) is useful for repeating over a data set to render table cells, and the [data grid](#) is useful for rendering complex data sets. In the simple cases of displaying read-only data, the repeater is similar to the JSTL `forEach` tag. In complex read / write cases, the repeater can participate with the other NetUI tags to create UI for editing entire data sets. For example, the repeater can be used to render HTML for editing quantities of items in a shopping cart.

2.1. Read only Data

The repeater binds to a data set using its `dataSource` attribute. The simplest repeater would render a TODO list as an unordered list like:

```
<%@ taglib uri="http://beehive.apache.org/netui/tags-databinding-1.0"
prefix="netui-data"%>
<ul>
<netui-data:repeater dataSource="pageSource.todoList">
  <li>${container.item.taskName} -- ${container.item.description}</li>
</netui-data:repeater>
</ul>
```

The container implicit object is available for data binding with the JSP 2.0 Expression Language and contains several properties including `index` and `item` which provide access to the index of the current item in the data set and to the current data item, respectively.

A header and footer can be added to the repeater by using the `repeaterHeader` and the `repeaterFooter` tags. For example, the above unordered list could be encapsulated in the repeater as:

```
<%@ taglib uri="http://beehive.apache.org/netui/tags-databinding-1.0"
prefix="netui-data"%>
<netui-data:repeater dataSource="pageSource.todoList">
  <netui-data:repeaterHeader>
    <ul>
      <netui-data:repeaterHeader>
      <netui-data:repeaterItem>
        <li>${container.item.taskName} --
${container.item.description}</li>
      <netui-data:repeaterItem>
      <netui-data:repeaterFooter>
    </ul>
    <netui-data:repeaterFooter>
  </netui-data:repeater>
```

The repeater can also display a default message when no data is available by setting the `defaultText` attribute. Additionally, when rendering sparse data sets, it is often useful to ignore null members; this can be enabled by setting the `ignoreNulls` attribute to `true`.

2.2. Read / write Data

The repeater tags can also be used to render read / write HTML markup. A simple example is displaying a user interface to change the quantity of each item in a shopping cart. The `samples/netui-samples/ui/repeaterediting/` sample in a Beehive distribution contains such an example. In this sample, the repeater appears as:

```
<netui:form action="update">
<netui-data:repeater dataSource="actionForm.items">
<tr>
  <td>${container.item.name}</td>
  <td>
    <netui:span value="${container.item.price}">
      <netui:formatNumber pattern="$####.00"/>
    </netui:span>
  </td>
  <td>
    <netui:textBox dataSource="container.item.quantity" size="5"/><br/>
    <font size="2" color="#ee0000"><netui:error
key="invalidQuantity${container.index}"/></font>
  </td>
  <td>
    <netui:checkBox dataSource="container.item.giftWrap"/>
  </td>
</tr>
</netui-data:repeater>
</netui:form>
```

In this sample, the repeater binds to an array of `Item` objects with a set of simple JavaBean properties `price`, `quantity`, and `giftWrap`. The `quantity` and `giftWrap` properties are rendered using the `<netui:textBox>` tag and `<netui:checkBox>` tags

which allow the browser to edit the values of those properties when the page POSTs.

This is possible because the NetUI form element tags work in cooperation with the repeater to rewrite expressions such as `container.item.giftWrap` into an expression that is used for the name of a check box in the form `actionForm.items[1].giftWrap`. The value of `container.index` for each item in the array is used to index into the `actionForm.items` array, and the property `giftWrap` is added to the end to form the new expression. The result is an expression that can POST to the server and can be used to look-up a specific form property on a specific item in the `Item` `JavaBean` array.

3. Cell Repeater

The cell repeater is a JSP tag that is used to render the contents of each cell in an HTML table. This user interface pattern is common in catalogs and shopping carts and is often used to display a single item per cell. The cell repeater can be constrained to render a table with a fixed length, a fixed width, or both a fixed length and width. For example, the following sample will display a table of `PetBean` objects, one per cell in a table two cells wide:

```
<netui-data:cellRepeater dataSource="pageScope.pets" columns="2">
  ID: <netui:span value="{container.item.petId}"/><br/>
  Name: <netui:span value="{container.item.name}"/><br/>
  Description: <netui:span value="{container.item.description}"/><br/>
  Price: <netui:span value="{container.item.price}"/><br/>
</netui-data:cellRepeater>
```