ICEmobile : DataView

This page last changed on Jun 07, 2013 by nils.lundquist.

DataView

Information

The DataView component is designed to enable lightweight viewing and editing of object collections. The component renders a table region for iterative output using columns defined with the required DataViewColumns component child. Also rendered is a detail region for in-depth viewing (and editing) of the row objects of the table model. The detail region is defined with the DataViewDetails child component and it's contents are displayed when a row of table region is activated, typically by a row tap.

The most important attributes of the `<mobi:dataview>` are comparable to those on a component like h:dataTable:

---

value - Bind a List or DataViewLazyDataModel to use as the iterative object collection
var - Define the EL variable to expose the individual iterative objects with during iteration

---

In addition to these attributes, there is another that particularly important attribute to the functioning of the details region.

---

activationMode - This enumeration defines the operation of the detail region. When set to 'server' (the default) the detail region may contain arbitrary JSF components and is rendered by an ajax request. When set to 'client' an activation 'renders' the detail region by updating, entirely on the client, an existing rendering with the dynamic attributes unique to an iterative rendering. The components and attributes supported for client encoding are limited, though growing with new releases and user desires.

The components and attributes currently supported for client activationMode alteration are:

Component	Attributes
JSF h: taglib UIInput Components	value
JSF h: taglib UIOutput Components	value
JSF h: taglib UICommand Components	value
mobi:inputText	value, type, placeholder, readonly, maxlength, size, required, results, title, min, max, step, disable, style, styleClass

Usage:

```
<h:form>
  <mobi:dataView value="#{dataViewBean.cars}" var="car">
    <mobi:dataViewColumns>
      <mobi:dataViewColumn headerText="Company" value="#{car.company}" />
      <mobi:dataViewColumn headerText="Model" value="#{car.model}" />
    </mobi:dataViewColumns>

    <mobi:dataViewDetails>
      <h3>Details</h3>
      <h:panelGrid columns="2">
        <h:outputText value="Company" />
```

```
                <h:outputText value="Model" />

                <h:inputText value="#{car.company}">
                    <f:ajax execute="@this" render="@form" />
                </h:inputText>
                <h:inputText value="#{car.model}">
                    <f:ajax execute="@this" render="@form" />
                </h:inputText>
            </h:panelGrid>
        </mobi:dataViewDetails>
    </mobi:dataView>
</h:form>
```

The example above is a simple use case of the DataView to show two columns of text information, and have the components to edit that information rendered in the detail view.

To see in detail how this example works refer to the tutorial here.

> ✅ **Documentation**
> This section covers attributes involved in the typical use-cases for this component. The complete taglib documentation for this component is available here.

Tag Handler Documentation

dataView