

Binding

Method Summary

void	setCallBackInfo (f, returnDataprovider) Set callback method for node selection and double click
void	setCheckBoxValueDataprovider (checkBoxValueDataprovider) Set checkbox value dataprovider.
void	setChildSortDataprovider (childSortDataprovider) Set the dataprovider name to retrieve column name and sort order for the child nodes.
void	setConfigurationDataprovider (configurationDataprovider) Set configuration dataprovider.
void	setFontTypeDataprovider (fontTypeDataprovider) Set the dataprovider name to retrieve the node font from
void	setHasCheckBoxDataprovider (hasCheckBoxDataprovider) Set has checkbox flag dataprovider.
void	setImageMediaDataprovider (imageMediaDataprovider) Set the dataprovider name to retrieve the node image from (blob column)
void	setImageURLDataprovider (imageURLDataprovider) Set the dataprovider name to retrieve the node image from (via url)
void	setMRelationDataprovider (dataprovider) Set m-relation dataprovider.
void	setMRelationName (name) Set m-relation name
void	setMethodToCallOnCheckBoxChange (methodToCallOnCheckBoxChange, returnDataproviderOnCheckBoxChange) Set method to call on check box status change
void	setMethodToCallOnClick (methodToCallOnClick, returnDataproviderOnClick) Set method to call on click.
void	setMethodToCallOnDoubleClick (methodToCallOnDoubleClick, returnDataproviderOnDoubleClick) Set method to call on double click.
void	setMethodToCallOnRightClick (methodToCallOnRightClick, returnDataproviderOnRightClick) Set method to call on right click.
void	setNRelationDataprovider (dataprovider) Set n-relation dataprovider.
void	setNRelationInfos (relationInfos) Set n-relation infos (array of RelationInfo objects created using tree.
void	setNRelationName (name) Set n-relation name
void	setTextDataprovider (textDataprovider) Set text dataprovider
void	setToolTipTextDataprovider (toolTipTextDataprovider) Set the dataprovider name to retrieve the node tooltip text from

Method Details

setCallBackInfo

void **setCallBackInfo** (f, returnDataprovider)
Set callback method for node selection and double click

Parameters

{[Function](#)} f
{[String](#)} returnDataprovider

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), controller.getTableName());
binding.setCallBackInfo(globals.node_selected, 'node_id');
```

setCheckBoxValueDataprovider

void **setCheckBoxValueDataprovider** (checkBoxValueDataprovider)
Set checkbox value dataprovider. Dataprovider returns INTEGER (0 or 1) or STRING (false or true)

Parameters

{String} checkBoxValueDataProvider

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), controller.getTableName());
binding.setCheckBoxValueDataProvider('comment_text');
```

setChildSortDataProvider

void **setChildSortDataProvider** (childSortDataProvider)

Set the dataprovider name to retrieve column name and sort order for the child nodes.\n\nThe provided data must be a string of form :
column_name_used_for_sort sort_order(asc or desc)

Parameters

{String} childSortDataProvider

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), 'customers');
binding.setChildSortDataProvider('company_sort');
```

setConfigurationDataProvider

void **setConfigurationDataProvider** (configurationDataProvider)

Set configuration dataprovider. Datasource must be MEDIA type and returns a configuration object

Parameters

{String} configurationDataProvider

Returns

void

Sample

```
var config = new Object();
config.text = 'my_text';
config.nRelation = 'my_n_relation';
config.mRelation = 'my_m_relation';
config.childSort = 'my_sort_field';
config.fontType = 'my_font';
config.toolTipText = 'my_tooltip';
config.hasCheckBox = 'true';
config.checkBoxValue = 'true';
```

setFontTypeDataProvider

void **setFontTypeDataProvider** (fontTypeDataProvider)

Set the dataprovider name to retrieve the node font from

Parameters

{String} fontTypeDataProvider

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), controller.getTableName());
binding.setFontTypeDataProvider('bean_font');
```

setHasCheckBoxDataProvider

void **setHasCheckBoxDataProvider** (hasCheckBoxDataProvider)

Set has checkbox flag dataprovider. Dataprovider returns INTEGER (0 / 1 / 2) or STRING (false / true / disabled) for (does not have / have / have but disabled)

Parameters

{String} hasCheckBoxDataprovider

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), controller.getTableName());
binding.setHasCheckBoxDataprovider('hasCheckBox');
```

setImageMediaDataprovider

void **setImageMediaDataprovider** (imageMediaDataprovider)

Set the dataprovider name to retrieve the node image from (blob column)

Parameters

{String} imageMediaDataprovider

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), 'customers');
binding.setImageMediaDataprovider('company_icon');
```

setImageURLDataprovider

void **setImageURLDataprovider** (imageURLDataprovider)

Set the dataprovider name to retrieve the node image from (via url)

Parameters

{String} imageURLDataprovider

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), 'orders');
binding.setImageURLDataprovider('order_icon');
```

setMRelationDataprovider

void **setMRelationDataprovider** (dataprovider)

Set m-relation dataprovider. Dataprovider returns the name of the m-relation

Parameters

{String} dataprovider

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), controller.getTableName());
binding.setMRelationDataprovider('m_relation');
```

setMRelationName

void **setMRelationName** (name)

Set m-relation name

Parameters

{String} name

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), controller.getTableName());
binding.setMRelationName('m_relation_name');
```

setMethodToCallOnCheckBoxChange

void **setMethodToCallOnCheckBoxChange** (methodToCallOnCheckBoxChange, returnDataproviderOnCheckBoxChange)

Set method to call on check box status change

Parameters

{Function} methodToCallOnCheckBoxChange
{String} returnDataproviderOnCheckBoxChange

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), 'orders');
binding.setMethodToCallOnCheckBoxChange(onCheckBoxChangeMethod, 'orderdate');
```

setMethodToCallOnClick

void **setMethodToCallOnClick** (methodToCallOnClick, returnDataproviderOnClick)

Set method to call on click.\n\nThe callback will be called with the following arguments : returnDataprovider, tableName, mouseX, mouseY

Parameters

{Function} methodToCallOnClick
{String} returnDataproviderOnClick

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), 'orders');
binding.setMethodToCallOnClick(onClickMethod, 'orderid');
```

setMethodToCallOnDoubleClick

void **setMethodToCallOnDoubleClick** (methodToCallOnDoubleClick, returnDataproviderOnDoubleClick)

Set method to call on double click.\n\nThe callback will be called with the following arguments : returnDataprovider, tableName, mouseX, mouseY

Parameters

{Function} methodToCallOnDoubleClick
{String} returnDataproviderOnDoubleClick

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), 'orders');
binding.setMethodToCallOnDoubleClick(doubleClickMethod, 'orderid');
```

setMethodToCallOnRightClick

void **setMethodToCallOnRightClick** (methodToCallOnRightClick, returnDataproviderOnRightClick)

Set method to call on right click.\n\nThe callback will be called with the following arguments : returnDataprovider, tableName, mouseX, mouseY

Parameters

{Function} methodToCallOnRightClick
{String} returnDataproviderOnRightClick

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), 'orders');
binding.setMethodToCallOnRightClick(rightClickMethod, 'orderid');
```

setNRelationDataprovider

void **setNRelationDataprovider** (dataprovder)

Set n-relation dataprovder. Dataprovider returns the name of the n-relation

Parameters

{[String](#)} dataprovder

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), controller.getTableName());
binding.setNRelationDataprovider('n_relation');
```

setNRelationInfos

void **setNRelationInfos** (relationInfos)

Set n-relation infos (array of RelationInfo objects created using tree.createRelationInfo() for having multiple child relations for one node)

Parameters

{[RelationInfo](#)[]} relationInfos

Returns

void

Sample

```
var company_relations = new Array();
company_relations[0] = tree.createRelationInfo();
company_relations[0].setLabel('Employees');
company_relations[0].setNRelationName('companies_to_employees');
company_relations[1] = tree.createRelationInfo();
company_relations[1].setLabel('Customers');
company_relations[1].setNRelationName('companies_to_customers');
binding.setNRelationInfos(company_relations);
```

setNRelationName

void **setNRelationName** (name)

Set n-relation name

Parameters

{[String](#)} name

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), controller.getTableName());
binding.setNRelationName('n_relation_name');
```

setTextDataprovider

void **setTextDataprovider** (textDataprovider)

Set text dataprovder

Parameters

{[String](#)} textDataprovider

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), 'orders');
binding.setTextDataprovider('orderid');
```

setToolTipTextDataprovider

void **setToolTipTextDataprovider** (toolTipTextDataprovider)

Set the dataprovider name to retrieve the node tooltip text from

Parameters

{[String](#)} toolTipTextDataprovider

Returns

void

Sample

```
var binding = elements.dbtreeview.createBinding(controller.getServerName(), 'customers');
binding.setToolTipTextDataprovider('companyname');
```