controller

Nov 21, 2019 22:44

Supported Clients

SmartClient WebClient NGClient MobileClient

Property Summary

Boolean	enabled	Gets or sets the enabled state of a form; also known as "grayed-out".
Boolean	readOnly	Gets or sets the read-only state of a form; also known as "editable" Note: The field(s) in a form set as read-only can be selected and the field data can be copied to clipboard.
Number	view	Get/Set the current type of view of this form.

Methods Summary

Boolean	deleteAllRecords()	Deletes all records in foundset, resulting in empty foundset.
Boolean	deleteRecord()	Delete current selected record, deletes mulitple selected records incase the foundset is using multiselect.
Boolean	duplicateRecord()	Duplicate current record or record at index in the form foundset.
Boolean	duplicateRecord(location)	Duplicate current record or record at index in the form foundset.
Boolean	duplicateRecord(location)	Duplicate current record or record at index in the form foundset.
Boolean	find()	Set the foundset in find mode.
void	focusField(fieldName, skipReadonly)	Sets focus to a field specified by its name.
void	focusFirstField()	Sets focus to the first field of the form; based on tab order sequence.
Number	getDataProviderMaxLength(name)	Returns the maximum length allowed in the specified dataprovider.
Object	getDataProviderValue(dataProvider)	Gets a value based on the specified dataprovider name.
String	getDataSource()	Get the used datasource.
Boolean	getDesignMode()	Returns the state of this form designmode.
Object	getDesignTimeProperty(key)	Get a design-time property of a form.
JSDataSet	getFormContext()	Gets the forms context where it resides, returns a dataset of its structure to the main controller.
Number	getFormWidth()	Gets the form width in pixels.
Number	getMaxRecordIndex()	Returns the current cached record count of the current foundset.
String	getName()	Get the name of this form.
Number	getPartHeight(partType)	Gets the part height in pixels.
Number	getPartYOffset(partType)	Returns the Y offset of a given part of the form.
Number	getSelectedIndex()	Gets the current record index of the current foundset.
Array	getTabSequence()	Get an array with the names of the components that are part of the tab sequence.
JSWindow	getWindow()	Returns the JSWindow that the form is shown in, or null if the form is not currently showing in a window.
Boolean	invertRecords()	Inverts the current foundset against all rows of the current table; all records that are not in the foundset will become the current foundset.
Boolean	loadAllRecords()	Loads all accessible records from the datasource into the form foundset.
Boolean	loadOmittedRecords()	Loads the records that are currently omitted in the form foundset.
Boolean	loadRecords()	Loads all accessible records from the datasource into the form foundset.
Boolean	loadRecords(foundset)	Loads a (related) foundset into the form.
Boolean	loadRecords(pkdataset)	Loads a primary key dataset, will remove related sort.
Boolean	loadRecords(foundset)	Loads a (related) foundset into the form.
Boolean	loadRecords(UUIDpk)	Loads a single record by primary key, will remove related sort.
Boolean	loadRecords(singlenNmber_pk)	Loads a single record by primary key, will remove related sort.
Boolean	loadRecords(queryString)	Loads records into form foundset based on a query (also known as 'Form by query').
Boolean	loadRecords(queryString, queryArgumentsArray)	Loads records into form foundset based on a query (also known as 'Form by query').
Boolean	newRecord()	Create a new record in the form foundset.
Boolean	newRecord(insertOnTop)	Create a new record in the form foundset.
Boolean	newRecord(location)	Create a new record in the form foundset.
Boolean	omitRecord()	Omit selected record(s) (add it to omit records list) in form foundset, to be shown with loadOmittedRecords.
void	print()	Print this form with current foundset, without preview.
void	print(printCurrentRecordOnly)	Print this form with current foundset, without preview.
void	print(printCurrentRecordOnly, showPrinterSelectDialog)	Print this form with current foundset, without preview.
void	print(printCurrentRecordOnly, showPrinterSelectDialog, printerJob)	Print this form with current foundset, without preview.
String	printXML()	Print this form with current foundset records to xml format.
String	printXML(printCurrentRecordOnly)	Print this form with current foundset records to xml format.
Boolean	recreateUI()	Recreates the forms UI components, to reflect the latest solution model.
void	relookup()	Performs a relookup for the current foundset record dataproviders.

Number	search()	Start the database search and use the results, returns the number of records, make sure you did "find" function first.
Number	search(clearLastResults)	Start the database search and use the results, returns the number of records, make sure you did "find" function first.
Number	search(clearLastResults, reduceSearch)	Start the database search and use the results, returns the number of records, make sure you did "find" function first.
void	setDataProviderValue(dataprovider, value)	Sets the value based on a specified dataprovider name.
void	setDesignMode(designMode)	Sets this form in designmode with param true, false will return to normal browse/edit mode.
void	setDesignMode(ondrag)	Sets this form in designmode with one or more callback methods.
void	setDesignMode(ondrag, ondrop)	Sets this form in designmode with one or more callback methods.
void	setDesignMode(ondrag, ondrop, onselect)	Sets this form in designmode with one or more callback methods.
void	setDesignMode(ondrag, ondrop, onselect, onresize)	Sets this form in designmode with one or more callback methods.
void	setDesignMode(ondrag, ondrop, onselect, onresize, ondblclick)	Sets this form in designmode with one or more callback methods.
void	setDesignMode(ondrag, ondrop, onselect, onresize, ondblclick, onrightclick)	Sets this form in designmode with one or more callback methods.
void	setPageFormat(width, height, leftmargin, rightmargin, topmargin, bottommargin)	Set the page format to use when printing.
void	setPageFormat(width, height, leftmargin, rightmargin, topmargin, bottommargin, orientation)	Set the page format to use when printing.
void	setPageFormat(width, height, leftmargin, rightmargin, topmargin, bottommargin, orientation, units)	Set the page format to use when printing.
void	setPreferredPrinter(printerName)	Set the preferred printer name to use when printing.
void	setSelectedIndex(index)	Sets the current record index of the current foundset.
void	setTabSequence(arrayOfElements)	Set the tab order sequence programatically, by passing the elements references in a javascript array.
void	show()	Shows the form (makes the form visible) This function does not affect the form foundset in any way.
void	show(window)	Shows the form (makes the form visible) This function does not affect the form foundset in any way.
void	show(window)	Shows the form (makes the form visible) This function does not affect the form foundset in any way.
void	showPrintPreview()	Show this form in print preview.
void	showPrintPreview(printCurrentRecordOnly)	Show this form in print preview.
void	showPrintPreview(printCurrentRecordOnly, printerJob)	Show this form in print preview.
void	showPrintPreview(printCurrentRecordOnly, printerJob, zoomFactor)	Show this form in print preview.
void	showRecords(foundset)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(foundset, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(foundset, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(pkdataset)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(pkdataset, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(pkdataset, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(query)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(query, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(query, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(UUIDpk)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(UUIDpk, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(UUIDpk, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(singleNumber_pk)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(singleNumber_pk, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(singleNumber_pk, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(query)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

void	showRecords(query, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(query, argumentsArray)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(query, argumentsArray, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(query, argumentsArray, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	showRecords(query, window)	Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.
void	sort(sortString)	Sorts the form foundset based on the given sort string.
void	sort(sortString, defer)	Sorts the form foundset based on the given sort string.
void	sortDialog()	Show the sort dialog to the user a preselection sortString can be passed, to sort the form foundset.
void	sortDialog(sortString)	

Property Details

enabled

```
Gets or sets the enabled state of a form; also known as "grayed-out".

Notes:
-A disabled element(s) cannot be selected by clicking the form.
-The disabled "grayed" color is dependent on the LAF set in the Servoy Smart Client Application Preferences.
```

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
//gets the enabled state of the form
var state = controller.enabled;
//enables the form for input
controller.enabled = true;
```

readOnly

```
Gets or sets the read-only state of a form; also known as "editable"

Note: The field(s) in a form set as read-only can be selected and the field data can be copied to clipboard.
```

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
//gets the read-only state of the form
var state = controller.readOnly;
//sets the read-only state of the form
controller.readOnly = true
```

view

```
Get/Set the current type of view of this form.
```

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
//gets the type of view for this form
var view = controller.view;
//sets the form to Record view
controller.view = 0;//RECORD_VIEW
//sets the form to List view
controller.view = 1;//LIST_VIEW
```

Methods Details

deleteAllRecords()

Deletes all records in foundset, resulting in empty foundset.

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var success = controller.deleteAllRecords();
```

deleteRecord()

Delete current selected record, deletes mulitple selected records incase the foundset is using multiselect.

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var success = controller.deleteRecord();
```

duplicateRecord()

Duplicate current record or record at index in the form foundset.

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.duplicateRecord(); //duplicate the current record, adds on top
//controller.duplicateRecord(false); //duplicate the current record, adds at bottom
//controller.duplicateRecord(1,2); //duplicate the first record as second record
```

duplicateRecord(location)

Duplicate current record or record at index in the form foundset.

Parameters

Boolean location true adds the new record as the topmost record

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.duplicateRecord(); //duplicate the current record, adds on top
//controller.duplicateRecord(false); //duplicate the current record, adds at bottom
//controller.duplicateRecord(1,2); //duplicate the first record as second record
```

duplicateRecord(location)

Duplicate current record or record at index in the form foundset.

Parameters

Number location adds at specified index

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.duplicateRecord(); //duplicate the current record, adds on top
//controller.duplicateRecord(false); //duplicate the current record, adds at bottom
//controller.duplicateRecord(1,2); //duplicate the first record as second record
```

find()

```
Set the foundset in find mode. (Start a find request), use the "search" function to perform/exit the find.

Before going into find mode, all unsaved records will be saved in the database.
If this fails (due to validation failures or sql errors) or is not allowed (autosave off), the foundset will
not go into find mode.
Make sure the operator and the data (value) are part of the string passed to dataprovider (included inside a
pair of quotation marks).
Note: always make sure to check the result of the find() method.

When in find mode, columns can be assigned string expressions (including operators) that are evaluated as:
General:
      c1||c2    (condition1 or condition2)
      c|format  (apply format on condition like 'x|dd-MM-yyyy')
      !c        (not condition)
      #c        (modify condition, depends on column type)
      ^         (is null)
      ^=        (is null or empty)
      &lt;x     (less than value x)
      &gt;x     (greater than value x)
      &lt;=x    (less than or equals value x)
      &gt;=x    (greater than or equals value x)
      x...y     (between values x and y, including values)
      x         (equals value x)

 Number fields:
      =x       (equals value x)
      ^=       (is null or zero)

 Date fields:
      #c       (equals value x, entire day)
      now      (equals now, date and or time)
      //       (equals today)
      today    (equals today)

 Text fields:
      #c                (case insensitive condition)
      = x      (equals a space and 'x')
      ^=       (is null or empty)
      %x%      (contains 'x')
      %x_y%    (contains 'x' followed by any char and 'y')
      \%      (contains char '%')
      \_      (contains char '_')

Related columns can be assigned, they will result in related searches.
For example, "employees_to_department.location_id = headoffice" finds all employees in the specified location).

Searching on related aggregates is supported.
For example, "orders_to_details.total_amount = '&gt;1000'" finds all orders with total order details amount
more than 1000.

Arrays can be used for searching a number of values, this will result in an 'IN' condition that will be used in
the search.
The values are not restricted to strings but can be any type that matches the column type.
For example, "record.department_id = [1, 33, 99]"
```

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
if (foundset.find()) //find will fail if autosave is disabled and there are unsaved records
{
        columnTextDataProvider = 'a search value'
        // for numbers you have to make sure to format it correctly so that the decimal point is in your locales
notation (. or ,)
        columnNumberDataProvider = '>' + utils.numberFormat(anumber, '####.00');
        columnDateDataProvider = '31-12-2010|dd-MM-yyyy'
        foundset.search()
}
```

focusField(fieldName, skipReadonly)

```
Sets focus to a field specified by its name.
If the second parameter is set to true, then readonly fields will be skipped
(the focus will be set to the first non-readonly field located after the field with the specified name; the tab
sequence is respected when searching for the non-readonly field).
```

Parameters

String fieldName the name of the field to be focussed
Boolean skipReadonly indication to skip read only fields, if the named field happens to be read only

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var tabseq = controller.getTabSequence();
if (tabseq.length > 1) {
        // If there is more than one field in the tab sequence,
        // focus the second one and skip over readonly fields.
        controller.focusField(tabseq[1], true);
}
else {
        // If there is at most one field in the tab sequence, then focus
        // whatever field is first, and don't bother to skip over readonly fields.
        controller.focusField(null, false);
}
```

focusFirstField()

Sets focus to the first field of the form; based on tab order sequence.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.focusFirstField();
```

getDataProviderMaxLength(name)

Returns the maximum length allowed in the specified dataprovider.

Parameters

String name the dataprovider name

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.getDataProviderMaxLength('name');
```

getDataProviderValue(dataProvider)

Gets a value based on the specified dataprovider name.

Parameters

String dataProvider the dataprovider name to retieve the value for

Returns

Object

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var val = controller.getDataProviderValue('contact_name');
```

getDataSource()

Get the used datasource.

Returns

String

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var dataSource = controller.getDataSource();
```

getDesignMode()

Returns the state of this form designmode.

Returns

Boolean

Supported Clients

SmartClient,WebClient

Sample

```
var success = controller.getDesignMode();
```

getDesignTimeProperty(key)

Get a design-time property of a form.

Parameters

String key the property name

Returns

Object

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var prop = forms.orders.controller.getDesignTimeProperty('myprop')
```

getFormContext()

Gets the forms context where it resides, returns a dataset of its structure to the main controller.
Note1: can't be called in onload, because no context is yet available at this time.
Note2: tabindex is 1 (left) or 2 (right) for a SplitPane and 0 based for the other tabpanels; tabindex1based is the same as tabindex but is 1 based.

Returns

JSDataSet

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
//dataset columns: [containername(1),formname(2),tabpanel or beanname(3),tabname(4),tabindex(5),tabindex1based
(6)]
//dataset rows: mainform(1) -> parent(2)  -> current form(3) (when 3 forms deep)
/** @type {JSDataSet} */
var dataset = controller.getFormContext();
if (dataset.getMaxRowIndex() > 1)
{
        // form is in a tabpanel
        var parentFormName = dataset.getValue(1,2)
}
```

getFormWidth()

Gets the form width in pixels.

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var width = controller.getFormWidth();
```

getMaxRecordIndex()

Returns the current cached record count of the current foundset.
To return the full foundset count, use: databaseManager.getFoundSetCount(...)
Tip: get the the table count of all rows in a table, use: databaseManager.getTableCount(...)

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
for ( var i = 1 ; i <= controller.getMaxRecordIndex() ; i++ )
{
        controller.setSelectedIndex(i);
        //do some action per record
}
```

getName()

Get the name of this form.

Returns

String

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var formName = controller.getName();
```

getPartHeight(partType)

Gets the part height in pixels.

Parameters

Number partType The type of the part whose height will be returned.

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var height = controller.getPartHeight(JSPart.BODY);
```

getPartYOffset(partType)

Returns the Y offset of a given part of the form.

Parameters

Number partType The type of the part whose Y offset will be returned.

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var offset = controller.getPartYOffset(JSPart.BODY);
```

getSelectedIndex()

Gets the current record index of the current foundset.

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
//gets the current record index in the current foundset
var current = controller.getSelectedIndex();
//sets the next record in the foundset, will be reflected in UI
controller.setSelectedIndex(current+1);
```

getTabSequence()

Get an array with the names of the components that are part of the tab sequence.
The order of the names respects the order of the tab sequence.
Components that are not named will not appear in the returned array, although they may be in the tab sequence.

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var tabseq = controller.getTabSequence();
if (tabseq.length > 1) {
        // If there is more than one field in the tab sequence,
        // focus the second one and skip over readonly fields.
        controller.focusField(tabseq[1], true);
}
else {
        // If there is at most one field in the tab sequence, then focus
        // whatever field is first, and don't bother to skip over readonly fields.
        controller.focusField(null, false);
}
```

getWindow()

Returns the JSWindow that the form is shown in, or null if the form is not currently showing in a window.

Returns

JSWindow

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var currentWindow = controller.getWindow();
if (currentWindow != null) {
        currentWindow.title = 'We have a new title';
} else {
        currentWindow = application.createWindow("Window Name", JSWindow.WINDOW, null);
        currentWindow(650, 700, 450, 350);
        currentWindow = "Window Title";
        controller.show(currentWindow);
}
```

invertRecords()

Inverts the current foundset against all rows of the current table; all records that are not in the foundset will become the current foundset.

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.invertRecords();
```

loadAllRecords()

Loads all accessible records from the datasource into the form foundset.
When the form contains a related foundset it will be replaced by a default foundset on same datasource.

Notes:
-the default foundset is always limited by filters, if databaseManager.addFoundSetFilterParam function is used.
-typical use is loading the normal foundset again after form usage in a related tabpanel

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.loadAllRecords();
```

loadOmittedRecords()

Loads the records that are currently omitted in the form foundset.

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.loadOmittedRecords();
```

loadRecords()

Loads all accessible records from the datasource into the form foundset. Typical usage is loading records after search in related tabpanel.
The difference to loadAllRecords() is that related foundset will load related records.

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
//to reload all last (related) records again, if for example after a search in related tabpanel
controller.loadRecords();
```

loadRecords(foundset)

Loads a (related) foundset into the form.
The form will no longer share the default foundset with forms of the same datasource, use loadAllRecords to restore the default foundset.

This will really update the foundset instance itself of the form, so now existing foundset is altered just the new foundset is shown.
This is different then doing foundset.loadRecords(foundset) because that just alters the current foundset and doesn't do anything with the foundset
that is given.

When the form uses a seperate foundset, foundset filter params are copied over from the source foundset and are merged with the existing filters.

Parameters

JSFoundSet foundset to load

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
//to load a (related)foundset into the form.
//the form will no longer share the default foundset with forms of the same datasource, use loadAllRecords to
restore the default foundset
controller.loadRecords(order_to_orderdetails);
```

loadRecords(pkdataset)

Loads a primary key dataset, will remove related sort. Tries to preserve selection based on primary key, otherwise first record is selected.

Parameters

JSDataSet pkdataset to load

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
//to load a primary key dataset, will remove related sort
//var dataset = databaseManager.getDataSetByQuery(...);
// dataset must match the table primary key columns (alphabetically ordered)
controller.loadRecords(dataset);
```

loadRecords(foundset)

Loads a (related) foundset into the form.
The form will no longer share the default foundset with forms of the same datasource, use loadAllRecords to restore the default foundset.

This will really update the foundset instance itself of the form, so now existing foundset is altered just the new foundset is shown.
This is different then doing foundset.loadRecords(foundset) because that just alters the current foundset and doesn't do anything with the foundset
that is given.

When the form uses a seperate foundset, foundset filter params are copied over from the source foundset and are merged with the existing filters.

Parameters

JSFoundSet foundset to load

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
//to load a (related)foundset into the form.
//the form will no longer share the default foundset with forms of the same datasource, use loadAllRecords to
restore the default foundset
controller.loadRecords(order_to_orderdetails);
```

loadRecords(UUIDpk)

Loads a single record by primary key, will remove related sort.

NOTE: This function will return true if the foundset was altered/changed. It is up to the developer to check for the presence of actual data using getSize().

Parameters

UUID UUIDpk to load

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.loadRecords(application.getUUID('6b5e2f5d-047e-45b3-80ee-3a32267b1f20'));
```

loadRecords(singlenNmber_pk)

Loads a single record by primary key, will remove related sort.

NOTE: This function will return true if the foundset was altered/changed. It is up to the developer to check for the presence of actual data using getSize().

Parameters

Number singlenNmber_pk to load

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.loadRecords(123);
```

loadRecords(queryString)

Loads records into form foundset based on a query (also known as 'Form by query'). The query must be a valid sql select.
If the foundset is related this function is not allowed.
Tries to preserve selection based on primary key, otherwise first record is selected.

see foundset.loadRecords(QBSelect).

When possible, the foundset will be loaded with the given query.
This is not always possible because the foundset needs to manipulate the query when adding conditions and joins.
In that case the query will be wrapped: select pk from tab where pk = (queryString)
The result is the same, except for the ordering in the queryString which will be ignored.

The query will be wrapped when one of the following is true:

you have no order-by clause
you have no from keyword
your query is not fully qualified on the main table
you have a group-by, having, join or union keyword

Parameters

String queryString to load

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.loadRecords(sqlstring);
```

loadRecords(queryString, queryArgumentsArray)

Loads records into form foundset based on a query (also known as 'Form by query'). The query must be a valid sql select.
If the foundset is related this function is not allowed.
Tries to preserve selection based on primary key, otherwise first record is selected.

see foundset.loadRecords(QBSelect).

When possible, the foundset will be loaded with the given query.
This is not always possible because the foundset needs to manipulate the query when adding conditions and joins.
In that case the query will be wrapped: select pk from tab where pk = (queryString)
The result is the same, except for the ordering in the queryString which will be ignored.

The query will be wrapped when one of the following is true:

you have no order-by clause
you have no from keyword
your query is not fully qualified on the main table
you have a group-by, having, join or union keyword

Parameters

String queryString to load
Array queryArgumentsArray the arguments to replace the questions marks in the queryString

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.loadRecords(sqlstring,parameters);
```

newRecord()

Create a new record in the form foundset.

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
// foreign key data is only filled in for equals (=) relation items
controller.newRecord();//default adds on top
//controller.newRecord(false); //adds at bottom
//controller.newRecord(2); //adds as second record
```

newRecord(insertOnTop)

Create a new record in the form foundset.

Parameters

Boolean insertOnTop true adds the new record as the topmost record

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
// foreign key data is only filled in for equals (=) relation items
controller.newRecord();//default adds on top
//controller.newRecord(false); //adds at bottom
//controller.newRecord(2); //adds as second record
```

newRecord(location)

Create a new record in the form foundset.

Parameters

Number location adds at specified index

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
// foreign key data is only filled in for equals (=) relation items
controller.newRecord();//default adds on top
//controller.newRecord(false); //adds at bottom
//controller.newRecord(2); //adds as second record
```

omitRecord()

Omit selected record(s) (add it to omit records list) in form foundset, to be shown with loadOmittedRecords.
This operation returns false only when form foundset is in bad state (table not accessible or not having a valid selected record)
or the record is in an edit state and can't be saved (autosave is false).

Note: The omitted records are discarded when these functions are executed: loadAllRecords, loadRecords (dataset), loadRecords(sqlstring), invert

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var success = controller.omitRecord();
```

print()

Print this form with current foundset, without preview.

Supported Clients

SmartClient,WebClient

Sample

```
//print this form (with foundset records)
controller.print();
//print only current record (no printerSelectDialog) to pdf plugin printer
//controller.print(true,false,plugins.pdf_output.getPDFPrinter('c:/temp/out.pdf'));
```

print(printCurrentRecordOnly)

Print this form with current foundset, without preview.

Parameters

Boolean printCurrentRecordOnly to print the current record only

Supported Clients

SmartClient,WebClient

Sample

```
//print this form (with foundset records)
controller.print();
//print only current record (no printerSelectDialog) to pdf plugin printer
//controller.print(true,false,plugins.pdf_output.getPDFPrinter('c:/temp/out.pdf'));
```

print(printCurrentRecordOnly, showPrinterSelectDialog)

Print this form with current foundset, without preview.

Parameters

Boolean printCurrentRecordOnly to print the current record only
Boolean showPrinterSelectDialog to show the printer select dialog (default printer is normally used)

Supported Clients

SmartClient,WebClient

Sample

```
//print this form (with foundset records)
controller.print();
//print only current record (no printerSelectDialog) to pdf plugin printer
//controller.print(true,false,plugins.pdf_output.getPDFPrinter('c:/temp/out.pdf'));
```

print(printCurrentRecordOnly, showPrinterSelectDialog, printerJob)

Print this form with current foundset, without preview.

Parameters

Boolean printCurrentRecordOnly to print the current record only
Boolean showPrinterSelectDialog to show the printer select dialog (default printer is normally used)
Object printerJob print to plugin printer job, see pdf printer plugin for example

Supported Clients

SmartClient,WebClient

Sample

```
//print this form (with foundset records)
controller.print();
//print only current record (no printerSelectDialog) to pdf plugin printer
//controller.print(true,false,plugins.pdf_output.getPDFPrinter('c:/temp/out.pdf'));
```

printXML()

Print this form with current foundset records to xml format.

Returns

String

Supported Clients

SmartClient,WebClient

Sample

```
//TIP: see also plugins.file.writeXMLFile(...)
var xml = controller.printXML();
//print only current record
//var xml = controller.printXML(true);
```

printXML(printCurrentRecordOnly)

Print this form with current foundset records to xml format.

Parameters

Boolean printCurrentRecordOnly to print the current record only

Returns

String

Supported Clients

SmartClient,WebClient

Sample

```
//TIP: see also plugins.file.writeXMLFile(...)
var xml = controller.printXML();
//print only current record
//var xml = controller.printXML(true);
```

recreateUI()

Recreates the forms UI components, to reflect the latest solution model.
Use this after altering the elements via solutionModel at the JSForm of this form.

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
// get the solution model JSForm
var form = solutionModel.getForm("myForm");
// get the JSField of the form
var field = form.getField("myField");
// alter the field
field.x = field.x + 10;
// recreate the runtime forms ui to reflect the changes.
controller.recreateUI();
```

relookup()

Performs a relookup for the current foundset record dataproviders.
Lookups are defined in the dataprovider (columns) auto-enter setting and are normally performed over a relation upon record creation.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.relookup();
```

search()

Start the database search and use the results, returns the number of records, make sure you did "find" function first.
Clear results from previous searches.

Note: Omitted records are automatically excluded when performing a search - meaning that the foundset result by default will not include omitted records.

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var recordCount = foundset.search();
//var recordCount = foundset.search(false,false); //to extend foundset
```

search(clearLastResults)

Start the database search and use the results, returns the number of records, make sure you did "find" function first.
Reduce results from previous searches.

Note: Omitted records are automatically excluded when performing a search - meaning that the foundset result by default will not include omitted records.

Parameters

Boolean clearLastResults boolean, clear previous search, default true

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var recordCount = foundset.search();
//var recordCount = foundset.search(false,false); //to extend foundset
```

search(clearLastResults, reduceSearch)

Start the database search and use the results, returns the number of records, make sure you did "find" function first.

Note: Omitted records are automatically excluded when performing a search - meaning that the foundset result by default will not include omitted records.

Parameters

Boolean clearLastResults boolean, clear previous search, default true
Boolean reduceSearch boolean, reduce (true) or extend (false) previous search results, default true

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var recordCount = foundset.search();
//var recordCount = foundset.search(false,false); //to extend foundset
```

setDataProviderValue(dataprovider, value)

Sets the value based on a specified dataprovider name.

Parameters

String dataprovider the dataprovider name to set the value for
Object value the value to set in the dataprovider

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.setDataProviderValue('contact_name','mycompany');
```

setDesignMode(designMode)

Sets this form in designmode with param true, false will return to normal browse/edit mode.

Parameters

Boolean designMode sets form in design mode if true, false ends design mode.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var form = forms["selectedFormName"];
if (!form.controller.getDesignMode())
{
        // Set the current form in designmode with no callbacks
        form.controller.setDesignMode(true);
        // Set the current form in designmode with callbacks
        // where onDrag, onDrop, onSelect, onResize are names of form methods (not from "selectedFormName" form)
        // form.controller.setDesignMode(onDrag, onDrop, onSelect, onResize);
}
//Set the current form out of designmode (to normal browse)
//form.controller.setDesignMode(false);
```

setDesignMode(ondrag)

Sets this form in designmode with one or more callback methods.

Parameters

Function ondrag onDrag method reference

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var form = forms["selectedFormName"];
if (!form.controller.getDesignMode())
{
        // Set the current form in designmode with no callbacks
        form.controller.setDesignMode(true);
        // Set the current form in designmode with callbacks
        // where onDrag, onDrop, onSelect, onResize are names of form methods (not from "selectedFormName" form)
        // form.controller.setDesignMode(onDrag, onDrop, onSelect, onResize);
}
//Set the current form out of designmode (to normal browse)
//form.controller.setDesignMode(false);
```

setDesignMode(ondrag, ondrop)

Sets this form in designmode with one or more callback methods.

Parameters

Function ondrag onDrag method reference
Function ondrop onDrop method reference

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var form = forms["selectedFormName"];
if (!form.controller.getDesignMode())
{
        // Set the current form in designmode with no callbacks
        form.controller.setDesignMode(true);
        // Set the current form in designmode with callbacks
        // where onDrag, onDrop, onSelect, onResize are names of form methods (not from "selectedFormName" form)
        // form.controller.setDesignMode(onDrag, onDrop, onSelect, onResize);
}
//Set the current form out of designmode (to normal browse)
//form.controller.setDesignMode(false);
```

setDesignMode(ondrag, ondrop, onselect)

Sets this form in designmode with one or more callback methods.

Parameters

Function ondrag onDrag method reference
Function ondrop onDrop method reference
Function onselect onSelect method reference

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var form = forms["selectedFormName"];
if (!form.controller.getDesignMode())
{
        // Set the current form in designmode with no callbacks
        form.controller.setDesignMode(true);
        // Set the current form in designmode with callbacks
        // where onDrag, onDrop, onSelect, onResize are names of form methods (not from "selectedFormName" form)
        // form.controller.setDesignMode(onDrag, onDrop, onSelect, onResize);
}
//Set the current form out of designmode (to normal browse)
//form.controller.setDesignMode(false);
```

setDesignMode(ondrag, ondrop, onselect, onresize)

Sets this form in designmode with one or more callback methods.

Parameters

- Function ondrag onDrag method reference
- Function ondrop onDrop method reference
- Function onselect onSelect method reference
- Function onresize onResize method reference

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var form = forms["selectedFormName"];
if (!form.controller.getDesignMode())
{
        // Set the current form in designmode with no callbacks
        form.controller.setDesignMode(true);
        // Set the current form in designmode with callbacks
        // where onDrag, onDrop, onSelect, onResize are names of form methods (not from "selectedFormName" form)
        // form.controller.setDesignMode(onDrag, onDrop, onSelect, onResize);
}
//Set the current form out of designmode (to normal browse)
//form.controller.setDesignMode(false);
```

setDesignMode(ondrag, ondrop, onselect, onresize, ondblclick)

Sets this form in designmode with one or more callback methods.

Parameters

- Function ondrag onDrag method reference
- Function ondrop onDrop method reference
- Function onselect onSelect method reference
- Function onresize onResize method reference
- Function ondblclick onDblClick method reference

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var form = forms["selectedFormName"];
if (!form.controller.getDesignMode())
{
        // Set the current form in designmode with no callbacks
        form.controller.setDesignMode(true);
        // Set the current form in designmode with callbacks
        // where onDrag, onDrop, onSelect, onResize are names of form methods (not from "selectedFormName" form)
        // form.controller.setDesignMode(onDrag, onDrop, onSelect, onResize);
}
//Set the current form out of designmode (to normal browse)
//form.controller.setDesignMode(false);
```

setDesignMode(ondrag, ondrop, onselect, onresize, ondblclick, onrightclick)

Sets this form in designmode with one or more callback methods.

Parameters

- Function ondrag onDrag method reference
- Function ondrop onDrop method reference
- Function onselect onSelect method reference
- Function onresize onResize method reference
- Function ondblclick onDblClick method reference
- Function onrightclick onRightClick method reference

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var form = forms["selectedFormName"];
if (!form.controller.getDesignMode())
{
        // Set the current form in designmode with no callbacks
        form.controller.setDesignMode(true);
        // Set the current form in designmode with callbacks
        // where onDrag, onDrop, onSelect, onResize are names of form methods (not from "selectedFormName" form)
        // form.controller.setDesignMode(onDrag, onDrop, onSelect, onResize);
}
//Set the current form out of designmode (to normal browse)
//form.controller.setDesignMode(false);
```

setPageFormat(width, height, leftmargin, rightmargin, topmargin, bottommargin)

Set the page format to use when printing.

Orientation values:
0 - Landscape mode
1 - Portrait mode

Units values:
0 - millimeters
1 - inches
2 - pixels

Note: The unit specified for width, height and all margins MUST be the same.

Parameters

Number	width	the specified width of the page to be printed.
Number	height	the specified height of the page to be printed.
Number	leftmargin	the specified left margin of the page to be printed.
Number	rightmargin	the specified right margin of the page to be printed.
Number	topmargin	the specified top margin of the page to be printed.
Number	bottommargin	the specified bottom margin of the page to be printed.

Supported Clients

SmartClient,WebClient

Sample

```
//Set page format to a custom size of 100x200 pixels with 10 pixel margins on all sides in portrait mode
controller.setPageFormat(100, 200, 10, 10, 10, 10);

//Set page format to a custom size of 100x200 pixels with 10 pixel margins on all sides in landscape mode
controller.setPageFormat(100, 200, 10, 10, 10, 10, SM_ORIENTATION.LANDSCAPE);

//Set page format to a custom size of 100x200 mm in landscape mode
controller.setPageFormat(100, 200, 0, 0, 0, 0, SM_ORIENTATION.LANDSCAPE, SM_UNITS.MM);

//Set page format to a custom size of 100x200 inch in portrait mode
controller.setPageFormat(100, 200, 0, 0, 0, 0, SM_ORIENTATION.PORTRAIT, SM_UNITS.INCH);
```

setPageFormat(width, height, leftmargin, rightmargin, topmargin, bottommargin, orientation)

Set the page format to use when printing.

Orientation values:
0 - Landscape mode
1 - Portrait mode

Units values:
0 - millimeters
1 - inches
2 - pixels

Note: The unit specified for width, height and all margins MUST be the same.

Parameters

Number	width	the specified width of the page to be printed.
Number	height	the specified height of the page to be printed.
Number	leftmargin	the specified left margin of the page to be printed.
Number	rightmargin	the specified right margin of the page to be printed.
Number	topmargin	the specified top margin of the page to be printed.
Number	bottommargin	the specified bottom margin of the page to be printed.
Number	orientation	the specified orientation of the page to be printed; the default is Portrait mode

Supported Clients

SmartClient,WebClient

Sample

```
//Set page format to a custom size of 100x200 pixels with 10 pixel margins on all sides in portrait mode
controller.setPageFormat(100, 200, 10, 10, 10, 10);

//Set page format to a custom size of 100x200 pixels with 10 pixel margins on all sides in landscape mode
controller.setPageFormat(100, 200, 10, 10, 10, 10, SM_ORIENTATION.LANDSCAPE);

//Set page format to a custom size of 100x200 mm in landscape mode
controller.setPageFormat(100, 200, 0, 0, 0, 0, SM_ORIENTATION.LANDSCAPE, SM_UNITS.MM);

//Set page format to a custom size of 100x200 inch in portrait mode
controller.setPageFormat(100, 200, 0, 0, 0, 0, SM_ORIENTATION.PORTRAIT, SM_UNITS.INCH);
```

setPageFormat(width, height, leftmargin, rightmargin, topmargin, bottommargin, orientation, units)

```
Set the page format to use when printing.

Orientation values:
0 - Landscape mode
1 - Portrait mode

Units values:
0 - millimeters
1 - inches
2 - pixels

Note: The unit specified for width, height and all margins MUST be the same.
```

Parameters

Number	width	the specified width of the page to be printed.
Number	height	the specified height of the page to be printed.
Number	leftmargin	the specified left margin of the page to be printed.
Number	rightmargin	the specified right margin of the page to be printed.
Number	topmargin	the specified top margin of the page to be printed.
Number	bottommargin	the specified bottom margin of the page to be printed.
Number	orientation	the specified orientation of the page to be printed; the default is Portrait mode
Number	units	the specified units for the width and height of the page to be printed; the default is pixels

Supported Clients

SmartClient,WebClient

Sample

```
//Set page format to a custom size of 100x200 pixels with 10 pixel margins on all sides in portrait mode
controller.setPageFormat(100, 200, 10, 10, 10, 10);

//Set page format to a custom size of 100x200 pixels with 10 pixel margins on all sides in landscape mode
controller.setPageFormat(100, 200, 10, 10, 10, 10, SM_ORIENTATION.LANDSCAPE);

//Set page format to a custom size of 100x200 mm in landscape mode
controller.setPageFormat(100, 200, 0, 0, 0, 0, SM_ORIENTATION.LANDSCAPE, SM_UNITS.MM);

//Set page format to a custom size of 100x200 inch in portrait mode
controller.setPageFormat(100, 200, 0, 0, 0, 0, SM_ORIENTATION.PORTRAIT, SM_UNITS.INCH);
```

setPreferredPrinter(printerName)

```
Set the preferred printer name to use when printing.
```

Parameters

String printerName The name of the printer to be used when printing.

Supported Clients

SmartClient,WebClient

Sample

```
controller.setPreferredPrinter('HP Laser 2200');
```

setSelectedIndex(index)

Sets the current record index of the current foundset.

Parameters

Number index the index to select

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
//gets the current record index in the current foundset
var current = controller.getSelectedIndex();
//sets the next record in the foundset, will be reflected in UI
controller.setSelectedIndex(current+1);
```

setTabSequence(arrayOfElements)

Set the tab order sequence programatically, by passing the elements references in a javascript array.

Parameters

Array arrayOfElements array containing the element references

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.setTabSequence([elements.fld_order_id, elements.fld_order_amount]);
```

show()

Shows the form (makes the form visible)
This function does not affect the form foundset in any way.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
// show the form in the current window/dialog
controller.show();
// show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.show(w);
// show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.show(w);
// or controller.show("mydialog");
//show the form in the main window
//controller.show(null);
```

show(window)

Shows the form (makes the form visible)
This function does not affect the form foundset in any way.

Parameters

JSWindow window the window in which this form should be shown, given as a window object

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
// show the form in the current window/dialog
controller.show();
// show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.show(w);
// show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.show(w);
// or controller.show("mydialog");
//show the form in the main window
//controller.show(null);
```

show(window)

Shows the form (makes the form visible)
This function does not affect the form foundset in any way.

Parameters

String window the window in which this form should be shown, specified by the name of an existing window

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
// show the form in the current window/dialog
controller.show();
// show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.show(w);
// show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.show(w);
// or controller.show("mydialog");
//show the form in the main window
//controller.show(null);
```

showPrintPreview()

Show this form in print preview.

Supported Clients

SmartClient,WebClient

Sample

```
//shows this form (with foundset records) in print preview
controller.showPrintPreview();
//to print preview current record only
//controller.showPrintPreview(true);
//to print preview current record only with 125% zoom factor;
//controller.showPrintPreview(true, null, 125);
```

showPrintPreview(printCurrentRecordOnly)

Show this form in print preview.

Parameters

Boolean printCurrentRecordOnly to print the current record only

Supported Clients

SmartClient,WebClient

Sample

```
//shows this form (with foundset records) in print preview
controller.showPrintPreview();
//to print preview current record only
//controller.showPrintPreview(true);
//to print preview current record only with 125% zoom factor;
//controller.showPrintPreview(true, null, 125);
```

showPrintPreview(printCurrentRecordOnly, printerJob)

Show this form in print preview.

Parameters

Boolean	printCurrentRecordOnly	to print the current record only
Object	printerJob	print to plugin printer job, see pdf printer plugin for example (incase print is used from printpreview)

Supported Clients

SmartClient,WebClient

Sample

```
//shows this form (with foundset records) in print preview
controller.showPrintPreview();
//to print preview current record only
//controller.showPrintPreview(true);
//to print preview current record only with 125% zoom factor;
//controller.showPrintPreview(true, null, 125);
```

showPrintPreview(printCurrentRecordOnly, printerJob, zoomFactor)

Show this form in print preview.

Parameters

Boolean	printCurrentRecordOnly	to print the current record only
Object	printerJob	print to plugin printer job, see pdf printer plugin for example (incase print is used from printpreview)
Number	zoomFactor	a specified number value from 10-400

Supported Clients

SmartClient,WebClient

Sample

```
//shows this form (with foundset records) in print preview
controller.showPrintPreview();
//to print preview current record only
//controller.showPrintPreview(true);
//to print preview current record only with 125% zoom factor;
//controller.showPrintPreview(true, null, 125);
```

showRecords(foundset)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

JSFoundSet	foundset	the foundset to load before showing the form.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(foundset, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

JSFoundSet foundset the foundset to load before showing the form.
JSWindow window the window in which this form should be shown, given as a window object.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(foundset, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

JSFoundSet foundset the foundset to load before showing the form.
String window the window in which this form should be shown, specified by the name of an existing window.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(pkdataset)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

JSDataSet pkdataset the pkdataset to load before showing the form.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(pkdataset, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

JSDataSet pkdataset the pkdataset to load before showing the form.
JSWindow window the window in which this form should be shown, given as a window object.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(pkdataset, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

JSDataSet pkdataset the pkdataset to load before showing the form.
String window the window in which this form should be shown, specified by the name of an existing window.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(query)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

QBSelect query the query to load before showing the form.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(query, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

QBSelect query the query to load before showing the form.
JSWindow window the window in which this form should be shown, given as a window object.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(query, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

QBSelect query the query to load before showing the form.
String window the window in which this form should be shown, specified by the name of an existing window.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(UUIDpk)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

UUID UUIDpk the UUIDpk to load before showing the form.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(UUIDpk, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

UUID UUIDpk the UUIDpk to load before showing the form.
JSWindow window the window in which this form should be shown, given as a window object.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(UUIDpk, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

UUID UUIDpk the UUIDpk to load before showing the form.
String window the window in which this form should be shown, specified by the name of an existing window.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(singleNumber_pk)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

Number singleNumber_pk the singleNumber_pk to load before showing the form.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(singleNumber_pk, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

Number singleNumber_pk the singleNumber_pk to load before showing the form.
JSWindow window the window in which this form should be shown, given as a window object

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(singleNumber_pk, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

Number singleNumber_pk the singleNumber_pk to load before showing the form.
String window the window in which this form should be shown, specified by the name of an existing window.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(query)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

String query the query to load before showing the form.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(query, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

String query the query to load before showing the form.
JSWindow window the window in which this form should be shown, given as a window object

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(query, argumentsArray)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

String	query	the query to load before showing the form.
Array	argumentsArray	the array of arguments for the query

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(query, argumentsArray, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

String	query	the query to load before showing the form.
Array	argumentsArray	the array of arguments for the query
JSWindow	window	the window in which this form should be shown, given as a window object

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(query, argumentsArray, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

String	query	the query to load before showing the form.
Array	argumentsArray	the array of arguments for the query
String	window	the window in which this form should be shown, specified by the name of an existing window.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

showRecords(query, window)

Load data into the form and shows the form, is a shortcut for the functions 'loadRecords' and 'show'.

Parameters

String query the query to load before showing the form.
String window the window in which this form should be shown, specified by the name of an existing window.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.showRecords(foundset);
// load foundset & show the form in newly created named modal dialog
var w = application.createWindow("mydialog", JSWindow.MODAL_DIALOG);
controller.showRecords(foundset, w);
// load foundset & show the form in an existing window/dialog
var w = application.getWindow("mydialog"); // use null name for main app. window
controller.showRecords(foundset, w);
//controller.showRecords(foundset, "mydialog");
```

sort(sortString)

Sorts the form foundset based on the given sort string.
Tries to preserve selection based on primary key. If first record is selected or cannot select old record it will select first record after sort.
TIP: You can use the Copy button in the developer Select Sorting Fields dialog to get the needed syntax string for the desired sort fields/order.

Parameters

String sortString the specified columns (and sort order)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.sort('columnA desc,columnB asc');
```

sort(sortString, defer)

Sorts the form foundset based on the given sort string.
Tries to preserve selection based on primary key. If first record is selected or cannot select old record it will select first record after sort.
TIP: You can use the Copy button in the developer Select Sorting Fields dialog to get the needed syntax string for the desired sort fields/order.

Parameters

String sortString the specified columns (and sort order)
Boolean defer the "sortString" will be just stored, without performing a query on the database (the actual sorting will be deferred until the next data loading action).

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.sort('columnA desc,columnB asc');
```

sortDialog()

```
Show the sort dialog to the user a preselection sortString can be passed, to sort the form foundset.
TIP: You can use the Copy button in the developer Select Sorting Fields dialog to get the needed syntax string
for the desired sort fields/order.
```

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.sortDialog('columnA desc,columnB asc');
```

sortDialog(sortString)

Parameters

String sortString the specified columns (and sort order)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
controller.sortDialog('columnA desc,columnB asc');
```