

Table of Contents

Appendix D. Links to Aware IM operations 2

Appendix D. Links to Aware IM operations

The following section lists the links to **Aware IM** operations that can be invoked from the hyperlinks of HTML pages included either in a visual perspective, custom menu or custom forms. All links have to call a particular JavaScript function supported by **Aware IM** in a special object called `AwareApp`, so the general format of a link is as follows:

```
<a href="#" onclick="AwareApp.functionName(parameters)" >Name of the link</a>
```

Many of these functions use “renderOption” parameter. This parameter indicates where the output of the function will be directed. Possible values are:

- `main`

Output will be displayed in the main page replacing the current contents

- `new_tab`

Output will be displayed in a new closable tab

- `popup`

Output will be displayed in a popup window

- Javascript object that defines properties of a modeless window
Output will be displayed in a modeless window. The object describing properties of a modeless window can have the following properties:
 - `modeless` (Boolean - must be defined and set to true)
 - `centered` (Boolean - if defined the window will be centered on the screen)
 - `position` - an object with two properties ('left' and 'top') that define where the window will appear (if `centered=false`)

Example:

```
{ modeless: true, position: { top: 100, left: 50 } }
```

- `AwareApp.getPanelId(frameType, tabName, panelName)`

Use this function to send the output of the operation to the specified panel of the specified tab in the specified frame, for example:

```
AwareApp.getPanelId ('main', 'MyTab', 'MyPanel')
```

- `#HTMLElementId`

If `renderOption` is a string that starts with the “#” symbol, the string after this symbol indicates a unique id of an HTML element where the output will be directed. The element can be anywhere on the screen. For example, if you have this placeholder element somewhere `<div id="my_id"></div>` and you specify `renderOption` as `#my_id`, the output of the operation will be displayed inside the placeholder element.

All supported JavaScript functions are presented in the table below:

Function	Example	Comments
startProcess (processName, renderOption)	AwareApp.startProcess ('MyProcess', 'main')	Start the process with the specified name
StartProcess2 (processName, ctxObjectName, ctxObjectId, renderOption)	AwareApp.startProcess2('MyProcess', 'MyObject', 256, 'main')	Start the process with the specified name and provide the specified object as a context
startProcessWithInit (processName, renderOption, initObjName, initObjParams, processContext)	AwareApp.startProcessWithInit('MyProcess', 'main', 'MyObject', 'Attr1=Value1@Attr2=Value2@...')	Create an object with the specified initialisation values and start a process that uses the created object as input. Initialisation values must be either a string in the form Attr1=Value1@Attr2=Value2@... etc where Attr1,Attr2 are attribute names in the object or it should be a JSON object with attribute values hashed by attribute names. If this function is used as firstCommand (see firstCommand) initialisation values will be appended automatically from the remaining URL parameters, for example: http://localhost:8080/AwareIM/LogonOp.aw?domain=B5&firstCommand=startProcessWithInit,MyProcess,main,MyObject&Attr1=Value1&Attr2=Value2...
startProcessFromForm (processName, renderOption, htmlElem, saveFlag)	AwareApp.startProcessFromForm ('MyProcess', 'main', this, true)	Usually used from a custom HTML element on an object form (button or hyperlink). HtmlElem parameter indicates this element. If saveFlag is true will save the form before starting a process. The saved object will be used as a parameter to the process
runQuery (queryName, renderOption)	AwareApp.runQuery ('My query', 'main')	Run the query with the specified name
runQueryWithParams (queryName, params, renderOption)	AwareApp.runQueryWithParams ('My query', params, 'main')	Run the query that requires user input. Provide the input values entered by the user in the params hashtable hashed by query parameter name
newObject (objectName, renderOption)	AwareApp.newObject('MyObject', 'main')	Display a form to create a new object with the specified name
getObjectData(objectName, objectId, callbackFunction)	AwareApp.getObjectData(context[0].objectName, context[0].objectId, function (objData) { alert(objData["Code"]); });	Asks the server to provide attribute values for the specified instance of the object(objectName and objectId). The callback function receives objData parameter that contains returned attribute values).
createOrUpdateObject (objectName, objectId, objectData, callbackFunction)	AwareApp.createOrUpdateObject ('A', null, { Txt: "abc", Nmb: 3 }, function (objectId, objectVersion) { alert ('Created' + objectId + " " + objectVersion); });	Calls the server to create or update the instance of the object with the specified data. The data is a JSON object with attribute names and values. The callback will receive the id and version of the object. If objectId is null a new instance will be created, otherwise an existing one will be updated.
saveForm (htmlElem)	AwareApp.saveForm (this)	Saves the current form. This can be used by custom buttons to perform save operation instead of the standard buttons. htmlElem defines an element on the form that performs the operation (button or hyperlink)
recalculateObject (htmlElem)	AwareApp.recalculateObject(this)	Similar to saveForm. However, if the form was displayed by a process, the process will continue displaying the form after the form has been saved.
recalculateObject2(htmlElem) - shortcut key on forms is F8	AwareApp.recalculateObject2(this)	Causes execution of rules of the object being edited or created using the values currently entered on the form. Displays recalculated values on the form
runDocument(docName)	AwareApp.runDocument ('My document template')	Run the document template with the specified name. The result is displayed in a new window
userDetails (renderOption)	AwareApp.userDetails ('main')	Display a form with the details of the currently logged in user
newUser(objectName, renderOption)	AwareApp.newUser ('MyUser', 'main')	Display a new user registration form for the user object with the specified name
showPerspective (perspective)	AwareApp.showPerspective ('MyPerspective')	Display the specified perspective
systemSettings (renderOption)	AwareApp.systemSettings ('main')	Display the form of the SystemSettings object
logout(logoutPage)	AwareApp.logout (null)	Logout from Aware IM. Optionally provide the URL of the logout page

From: <http://www.awareim.com/dokuwiki/> - **Documentation**

Permanent link: http://www.awareim.com/dokuwiki/docs/8000_appendices/0400_links_to_operations?rev=1660992954

Last update: **2022/09/13 18:11**

