JQXB – Expression Binder 1.1 MAIN APIs Cheat Sheet (build 2012.07.13)

Initialization

jQXB.initialize(verify,alertOnError)
Initialize the system.
verify = future use
alertOnError = future use

jQXB.compatibilitymode
default = false
Set this property to true for compatibility with jQuery's version prior to 1.6

Data sources operations

Creation
jQXB.setDataSource(name,object[s][,autorefresh])
create or set a datasource for binding.
i.e. single Object:
jQXB.setDataSource('mysource',{'Name':'Steve','Age':40});
i.e. Object List:
jQXB.setDataSource('mylist',[{'Name':'Steve,'Age':40},{'Name':'Al an','Age':35}]);
if [autorefresh] is set to true each change to the datasource will be immediately reflected in all HTML elements associated.

Start Binding
jQXB.doBind([name][,contextScope])
starts binding operations and fills associated html elements with appropriate values from datasources.
name = specific datasource for which execute binding process
contextScope = future use.
i.e.: jQXB.doBind('mysource');
If name parameter is not specified all registered datasource will be affected.

Retrieving
jQXB.getDataSource(name)
returns the Object[s] from the connected datasource,
i.e.: var MyObj = jQXB.getDataSource('mysource');

Orig Underlying Data
jQXB.getDataSourceOrigValue(name)
returns the original Object[s] passed in the last setDataSource invocation.
i.e.: var OldObj = jQXB.getDataSourceOrigValue('mysource');

Adding/ Removing Objects to/from datasource containing
jQXB.addObjectToDataSource(name,object)
append a new object to the datasource used as a template data source

I.e.:
jQXB.addObectToDataSource('mylist',{'Name':'Stevie','Age':35});
jQXB.deleteObjectFromDataSource(name,idxelem)
remove the idxelem from the datasource used as template datasource
I.e.: jQXB.deleteObjectFromDataSource('mylist',2)

Template operations

Removing
jQXB.clearTemplateInstances(templatename)
Remove all template items from page for a specific template
I.e.:
jQXB.clearTemplateInstances('myTemplate');
jQXB.deleteTemplateRow(templatename,idxrow)
Remove the idxrow template element from page
I.e.:
jQXB.deleteTemplateRow('myTemplate',2);

Events subscription

The following methods are used to subscribe to different platform events

jQXB.addOnBeforeDataSourceBindhnd(handler)
attachs an event handler callback invoked just before a datasource is binded to html elements (subscribe onBeforeDataSourceBind event)
jQXB.addOnBeforeUpdatehdn(handler)
attachs an event handler to be invoked before modification of the underlying datasource.(subscribe onBeforeUpdate event)
jQXB.addOnBeforeTemplateBindhnd(handler)
attachs an event handler to invoke before binding datasource to templates (subscribe onBeforeTemplateBind event)
jQXB.addOnAfterUpdatehnd(handler)
attachs an event handler to invoked after modification to the underlying data (subscribe onAfterUpdate event)
jQXB.addOnTemplateItemBoundhnd(handler)
attachs an event handler to invoke after the creation of a new template item (subscribe onTemplateItemBound event)
jQXB.addOnAfterDataSourceBindhnd(handler)
attachs an event handler to be invoked after the binding process for a datasource (subscribe onAfterDataSourceBind event)

Callbacks for events subscriptions

When a specific event occurs, the jQXB will notify all registered event handlers.

each handler will be passed parameters related to the context within which these events occurred.
To correctly handle messages that are sent from jQXB event handlers, the developer must define these operators according to a precise signature.

Event:
onBeforeDataSourceBind
Signature:
function(dataSourcename,jsobject)
dataSourceName = name used for jQXB.setDataSource method
jsobject = object stored

Event:
onBeforeUpdate
Signature:
function(dataSourcename,occurrency,member,membername e,value)
dataSourcename = name used for jQXB.setDataSource method
occurrency = [null] if object or index for element array
member = the object's member containing the current value to be changed
*membername = the name as declared into **jqxb-datamember** attributes.*
value = the new value.
This handler can also return true or false to specify a cancel condition in order to abort the value changing. (true = cancel)

Event:
onTemplateItemBound
Signature:
function(dataSourcename,templateName,occurrency,itemDa ta ,itemTemplate)
dataSourceName = name used for jQXB.setDataSource method
templateName = the name of the template container
occurrency = the idx of the element just binded
itemData = the idx-element just binded
itemTemplate =a jQuery object containing the HTML element belonging to the template

Event:
onBeforeTemplateBind
Signature:
function(dataSourcename,jsobject,templateName)
dataSourceName = name used for jQXB.setDataSource method
jsobject = the object in datasource that is about to be associated with the template

jQXB-M

JQXB – Expression Binder 1.1 MAIN APIs Cheat Sheet (build 2012.07.13)

templateName = the name of the template

Event:
onAfterUpdate
Signature:
function(dataSourcename,occurrency,member,membernam e,value)
dataSourcename = name used for jQXB.setDataSource method
occurrency = [null] if object or index for element array
member = the object's member containing value just changed
*membername = the name as declared into **jqxb-datamember** attributes.*
value = the new value.

Event:
onAfterDataSourceBind
Signature:
function(dataSourcename,jsobject)
dataSourceName = name used for jQXB.setDataSource method
jsobject = object stored

Remote data operation

JQXB exposes three main methods for retrieving and sending json object from/to server. These functions are essentially shortcuts for jQuery corresponding functions.
Retrieving Data
jQXB.getJSON(url,jsobject,successfunc(data){},failurefunc(x hr){})
retrieve a jSON object from server and passes handlers for success or failure (method GET)
jQXB.saveJSON(url,jsobject,successfunc(data){},failurefunc(xhr){})
send an object to the server using the POST verb
jQXB.deleteJSON(url,jsobject,successfunc(data){},failurefun c(xhr){})
send an object to the server using the DELETE verb

Miscellaneus

Refresh the data displayed using the actual values stored in the datasource.
jQXB.refreshControls(sourcename[,idx-elem][,memberame])

Reload data from datasource to HTML element. Idx-elem must be specified for a template datasource. If membername parameter is not specified JQXB refresh all HTML elements binded to the datasource, if specified JQXB refresh the HTML elements binded to the specific member.
jQXB.toJSON(value)
serializes an object into jSON format.

jQXB.utils.loadWidget(widgetUrl,targetElementId)
Loads a Widget specified by widgetUrl (Html page or a portion of HTML code) into an HTML element identified by its "id". The way loadWidget loads the widget allows the execution of any scripting contained in the Widget itself. This function helps the UI composite development when different parts of UI can be loaded conditionally or in different moments. The execution of Javascript contained into the loaded Widget assure the correct subscription of messages in case of jQXBM messaging subsystem.

JQXB – Expression Binder 1.1 MAIN APIs Cheat Sheet (build 2012.07.13)

HTML custom attributes

jQXB is declarative. That means you must specify
the guidelines for its proper functioning. Declarations
must be present within the page's HTML code using
a set of custom attributes for items we wish to
associate with data.

jqxb-datasource
used to specify the name of datasource to bind

jqxb-datamember
*used to specify the name of the member within the datasource to
bind to the current html element*

jqxb-templatecontainer
used to specify the unique name of the template

jqxb-template
used to specify the container belonging to

jqxb-templateitemidprfx
used to specify the prefix to apply for every new item template id

jqxb-itemdatamember
same as jqxb-datamember inside a template definition context

jqxb-bindedattribute
*used to specify the attribute to associate to datasource member
value if different from value (i.e.: width, padding, length)*

jqxb-transformfunc
*used to specify the function name of the value transformation
function*

HTML Lists/Combo attributes

jqxb-listsource
*used to specify the name of the datasource to use for populating
list data*

jqxb-listvalue
*used to specify which member of datasource to use as list
elements value*

jqxb-listtext
*used to specify which member of datasource to use as list
elements text*

jQXB-M