

User Interface - Feature #3258

improve javascript upcalls from the embedded client

03/07/2017 12:30 PM - Greg Shah

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>version:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			
<b>Related issues:</b>			
Related to User Interface - Feature #3218: enhance the embedded mode web GUI ...		<b>Closed</b>	
Related to User Interface - Feature #3259: add useful web components that can...		<b>New</b>	
Related to User Interface - Feature #3260: auto-conversion and enablement for...		<b>Closed</b>	
Related to User Interface - Feature #3261: enhanced browse that can optionall...		<b>Closed</b>	
Related to User Interface - Feature #3469: final improvements and polish for ...		<b>Closed</b>	

History

#1 - 03/07/2017 12:39 PM - Greg Shah

There are 3 high priority changes needed for the embedded web client's javascript upcalls:

1. To the degree that these calls are not fully secured on the Java side (integrated with our security manager infrastructure and plugins), we need to add this.
2. We need to move the parameter passing conversion and management back into the runtime. Instead of writing ABL code to consume or generate JSON, we should be automatically translating parameters (input, input-output and output) into a single JSON return object. Each parameter would have objects at the top level and it should be smart about how to convert each type, extents and so forth. We should handle the simple and extent BDT cases. But just as important is the temp-table case. The idea is that the ABL should not have to know that it was called by javascript, we should be able to handle the JSON creation completely in the runtime.
3. I also would like to make sure we do safety checking of all inputs and ensure that only safe values will be translated to ABL types. Any problems detected should cause the upcall to be rejected without abending.

#2 - 03/07/2017 12:39 PM - Greg Shah

- Related to Feature #3218: enhance the embedded mode web GUI to implement a good default approach for common cases added

#3 - 03/07/2017 12:43 PM - Greg Shah

Eric also had the idea about trying to make it easy to implement bidirectional data binding.

When using something like [D3](#), the toolset makes it easy to bind your UI to JSON data. It would be very powerful to enable birdirectional updates so that changes on the web side could be pushed back to the ABL easily.

**#4 - 03/07/2017 12:45 PM - Greg Shah**

*- Related to Feature #3259: add useful web components that can be easily integrated into the embedded web client added*

**#5 - 03/07/2017 12:54 PM - Greg Shah**

To provide the best solution for the auto-JSON creation, we would want to allow ABL OO class graphs and prodatasets to be handled cleanly. These both provide non-trivial data structuring options that are natural for ABL programmers. Fully enabling these will allow these data structures to be properly reflected down to the web client.

**#6 - 03/07/2017 01:19 PM - Greg Shah**

*- Related to Feature #3260: auto-conversion and enablement for the embedded web client added*

**#7 - 03/07/2017 01:24 PM - Greg Shah**

*- Related to Feature #3261: enhanced browse that can optionally selected as a replacement for the default ABL browse added*

**#8 - 03/05/2018 02:37 PM - Greg Shah**

*- Related to Feature #3469: final improvements and polish for the embedded mode web client added*