Question
Craig Regester · Jun 17, 2020

DTL Editor actions with persistent/serialized classes

Good day all -

I am attempting to use the Data Transformation Builder (for ease of use for my other engineers) to build up a web service request object to send to an outbound operation. The source is a custom persistent class (extends Ens.Response, %JSON.Adaptor) and has serialized sub-class data elements and the request object is a custom persistent class (extends Ens.Request, %JSON.Adaptor.)

When attempting to use a for each on one of the serialized properties of the source object, the for each works (source.Items()) but when I get into inspecting, via a conditional, the items within, it doesn't work like I thought it would. If I attempt to access an item like:

source.Items.(i).Value

It does not work. I have to use:

source.Items.GetAt(i).Value

And this works. But of course the DTL Editor doesn't project the GetAt function when defining the action automatically from the drop-down - I have to manually key it in each time (which is not very user friendly to my other engineers that may not know to do this intrinsically.) The DTL Editor projects the first statement, which doesn't work.

Why is this? Is there something I missed when defining my custom message classes that instructs the DTL Editor how to properly project or access the items within a list collection? Or is the DTL Editor simply not meant to work with these custom classes in the same way?

I can get it to work, obviously, but more just trying to understand from a learning perspective.

Thanks,

Craig

#DTL #Object Data Model #ObjectScript #Ensemble #InterSystems IRIS #InterSystems IRIS for Health