Announcement
Stefan Wittmann · Aug 25, 2016

JSON changes in Caché 2016.2

As Bill has mentioned earlier in his post, we have carefully reviewed the JSON capabilities and made some adjustments to ensure they deliver the best benefit to you. In this post, I am going to describe the modifications in more detail and provide guidance for you to understand the implication for your code base.

1) Class name changes

We felt the naming of %Object and %Array was too generic and that's why we are renaming them. %Object is now named %DynamicObject and %Array is now named %DynamicArray. If you are using the object initialization syntax (curly braces and square brackets) to initialize your objects and arrays you are unaffected by this change. If you have used the class name directly, for example:

```
set object = ##class(%Object).$new()
set array = ##class(%Array).$new()
set json = ##class(%Object).$fromJSON(someJSONstring)
```

use the new class name instead:

```
set dynamicObject = ##class(%DynamicObject).$new()
set dynamicArray = ##class(%DynamicArray).$new()
set json = ##class(%DynamicObject).$fromJSON(someJSONstring)
```

2) System methods

System methods have been completely removed from the product. All system methods that have been introduced in Caché 2016.1 are now present as regular methods starting with a leading "%" (percent) character followed by an uppercase letter. Here is a simple example: The above system class method $fromJSON has been converted to a regular class method named %FromJSON.

This is how the above code snippet has to be changed to work in Caché 2016.2:

```
set dynamicObject = ##class(%DynamicObject).%New()
set dynamicArray = ##class(%DynamicArray).%New()
set json = ##class(%DynamicObject).%FromJSON(someJSONstring)
```

3) Object initialization

The object initialization is one of the great additions to the JSON capabilities. While we worked with some of our customers we identified two pieces that required a revision. I want to make sure we are all talking about the same thing, so here is a little example of an object initialization for a dynamic object:

```
set myObject = {
```

```
"a string":"this is a ""quoted"" string",
"version":$zv,
"sum":1+2
}
```

The first change is that all value expressions have to be enclosed in parenthesis. There were some cases where it was ambiguous what the developer meant and we are correcting this. As a result, you have to enclose the values for the keys "version" and "sum" in parenthesis from now on.

The second change is that all literal values will be evaluated as JSON syntax and not as Caché Object Script literals. We want the object initialization to be a great tool to make it easy for you to create dynamic JSON content. The above example includes a string literal that is escaped by Caché Object Script syntax. From now on you have to escape a double quote with a backward slash. If you really want to use Caché Object Script escaping, you can enclose a string with parenthesis.

Here is the above example reworked for Caché 2016.2:

```
set myObject = {
"a string":"this is a \"quoted\" string",
"version":($zv),
"sum":(1+2)
}
```

Here is another alternative, enclosing the string with parenthesis to enforce Caché Object Script evaluation:

```
set myObject = {
"a string":("this is a ""quoted"" string"),
"version":($zv),
"sum":(1+2)
}
```

4) Composition

The method $compose has been removed from Caché 2016.2. We strongly believe that the functionality to convert between registered and dynamic objects is a critical piece to the JSON capabilities. Because of this, we want to make sure the API fits our customer needs and we will be reworking the $compose method to make it easier to use.

5) SQL changes

The JSONTABLE function has been removed from Caché 2016.2. This originates from the removal of the document data model. The other implemented JSON functions JSONOBJECT and JSONARRAY remain unimpacted.

Ben posted an article here, describing some guidelines we are using internally when you have to write forward compatible code in Caché 2016.1.

If you have questions or concerns regarding this announcement, please feel free to post them on the Developer Community or contact your Account Manager or Sales Engineer. You are also welcome to contact the Worldwide Response Center (Support.InterSystems.com).

#Compatibility #JSON #SQL #Caché

Source URL:https://community.intersystems.com/post/json-changes-cach%C3%A9-20162