Question
Josh Tennefoss · Dec 9, 2016

JSON, protect against large number truncation in javascript

How can I prevent large number truncation when using the DynamicObject and DynamicArray classes to create JSON in objectscript and then parsing the result in javascript. Are there any global settings that configure the output when using these classes?

I am looking at converting to the new DynamicObject and DynamicArray classes to build JSON. Our current (homegrown) JSON library forces string representation of numbers too long because of errors we were seeing with long numbers being truncated on the client, I am hoping to accomplish the same with the new classes.

```
USER>set longNum = 1234567890123456789

USER>w longNum
1234567890123456789 //valid cache number, 19 digits
USER>s arr = [(longNum)]

USER>w arr.%ToJSON()
[1234567890123456789] //Valid JSON, however I would like this to be a quoted string so it can be parsed in
javascript

//Parsed in javascript

JSON.parse('[1234567890123456789]')
[1234567890123456800] //Truncated, exceeds max integer
```

I know the issue is handled for any string type variable (which greatly lowers the chance of an error) or can be prevented by forcing the type for every single use, but if at all possible I would like to eliminate all chances of this error (which can be difficult to detect because it only presents for select data, and even then sometimes only when numeric casting has occurred for some reason).

```
USER>set strNum = "1234567890123456789" //declared as a string, not an issue

USER>s safeArr = [(strNum)].%Push(longNum, "string") //type set manually, but can't use the inline expression

USER>w safeArr.%ToJSON()
["1234567890123456789","1234567890123456789"] //JSON is safe to use in javascript (but again, only safe if you
remember to take the above steps or don't get an unexpectedly large number)
```

#.NET #Error Handling #JSON #ObjectScript #Caché