

Question

[Jeffrey Semmens](#) · May 3, 2016

Modifying REST.DocServer to use new %Object instead of %ZEN.proxyObject

I was trying to modify the REST.DocServer sample in the SAMPLE namespaces to use the new %Object and %Array system objects (Ensemble 2016.2 build 636) with the following changes:

```
/// This method returns a list of namespaces for this server
```

```
ClassMethod GetNamespaces() As %Status
```

```
{
```

```
    #dim tSC As %Status = $$$OK
```

```
    #dim tList,tNS,tFilteredList,tValue As %String
```

```
    #; Get the list of namespaces
```

```
    Do List^%SYS.NAMESPACE(.tList)
```

```
    Set tNS="" For {
```

```
        Set tNS=$Order(tList(tNS),1,tValue) If tNS="" Quit
```

```
        #; Only want namespaces which are enabled and are not remote
```

```
        If $List(tValue,1)=1&&$List(tValue,2)=0 {
```

```
            Set tFilteredList(tNS)=""
```

```
        }
```

```
    }
```

```
    #; If we have been asked for json
```

```
    If $Get(%request.CgiEnvs("HTTP_ACCEPT"))="application/json" {
```

```
        #; Set the response header to JSON
```

```
Set %response.ContentType="application/json"
```

```
##; Create a JSON proxy
```

```
set tProxy = {}
```

```
;Set tProxy = ##class(%ZEN.proxyObject).%New()
```

```
##; And a list to hold the namespaces
```

```
set tList= []
```

```
;Set tList=##class(%Library.ListOfDataTypes).%New()
```

```
##; Add the namespaces to the list
```

```
Set tNS="" For {
```

```
    Set tNS=$Order(tFilteredList(tNS)) If tNS="" Quit
```

```
    do tList.$push(tNS)
```

```
    ;Do tList.Insert(tNS)
```

```
}
```

```
##; Set the namespace property
```

```
Set tProxy.namespaces=tList
```

```
##; Output the JSON
```

```
set ^jks=tProxy.$toJSON()
```

```
do tProxy.$toJSON()
```

```
;Do tProxy.%ToJSON()
```

```
} else {
```

```
Set tNS="" For {
```

```
    Set tNS=$Order(tFilteredList(tNS)) If tNS="" Quit
```

```
Write tNS,!
```

```
}
```

```
}
```

```
Quit tSC
```

```
}
```

Unfortunately this returns an empty response when run. It LOOKS like both the original code do the same things but when I tested the original by setting a global equal to the value of `tProxy.%ToJSON()`, I got a 1 and the JSON sent twice in the response. When I set a global = `tProxy.$toJSON()` I actually get the JSON in the global. Is `%ZEN.proxyObject.%ToJSON()` doing something specific with the device that `$toJSON()` doesn't?

[#Document Data Model \(NoSQL\)](#) [#Field Tests](#) [#REST API](#) [#Caché](#)

Source URL: <https://community.intersystems.com/post/modifying-restdocserver-use-new-object-instead-zenproxyobject>