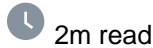


## Serialize and Deserialize objects to JSON (or whatever) format

Article

[Jose Tomas Salvador](#) · Apr 25, 2018



2m read

## Serialize and Deserialize objects to JSON (or whatever) format

What if you could serialize/deserialize objects in whatever format: JSON, XML, CSV,...; attending different criteria: export/import some properties and not others, transform values in this or that way before exporting/importing,...; and **all of this without having to change the class definition**? Wouldn't that be great??

Well, perhaps it's a goal too ambitious to reach 100% but, exploring this idea, I've developed a bunch of classes that I thought it was good to share. If you want to test, change, modify or improve the code, or just take a look at it, you can do it [here](#). There you'll find a more detailed explanation (see Readme.md)

**Be aware**, this is a proof of concept for myself done in spare times, sure it's not robust enough or it can be done much better... but, I was just playing...ok, I could just wait to the new JSON Adaptor (coming soon!) that sure is going to resolve much more scenarios in a cleaner way, but... meanwhile... :-)

Basically, you just have to import the classes. Then choose one of your persistent or registered classes or create a new one, and change it to extend `OPNLib.Serialize.Adaptor`. After that, you just can do the following:

```
set obj = ##class(MyPkg.MyPersistentClass).%OpenId(1)
set json = obj.Export()
do json.%ToJSON()
{"prop1":"value1","prop2":"value"}

set newObj = ##class(MyPkg.MyClass).%New()
do newObj.Import(json)
write newObj.%Save()
write newObj.%Id() //if it's a persistent object, of course
27
```

This approach doesn't entirely avoid the need to modify the class definition yet... but once you do it, you'll have 2 methods that can act as dispatchers for whatever other serialization logic you want to inject in the future... and also, if you explore the idea behind the `OPNLib.Serialize.Template` class, it'll be relatively easy to adapt it to have a pre-compiled export/import methods to export/import data in whatever serialization format without modifying the source/target class.

### Enjoy!

[#Code Snippet](#) [#JSON](#) [#Mapping](#) [#Object Data Model](#) [#REST API](#) [#Caché](#)

90 2 4 4 1,370

Log in or sign up to continue  
Add reply

**Source URL:** <https://community.intersystems.com/post/serialize-and-deserialize-objects-json-or-whatever-format>