classToJSON

Syntax

```
set aString = anObject.classToJSON()
set aString = anObject.classToJSON(options)
```

Semantics

The operation `classToJSON()` takes any object (`anObject`) and tries to map it to a JSON string `aString`. If this is not possible, an error is raised.
JSON types and xUML types are mapped as listed in chapter jsonToClass() Operation.

> `classToJSON()` does not support cyclic structures. When a cyclic structure is detected during serializing the given object, an exception will be thrown.
>
> Runtimes before 2023.1 Trying to serialize a cyclic structure leads to a service crash.

Builder 7.5.0 You can hide attributes from being serialized by applying stereotype <<E2EPrivate>>. Refer to Hiding Attributes From Interfaces for more information.

Substitutables

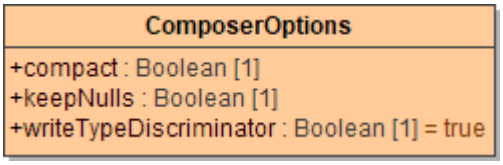
`anObject`	Any object.
`aString`	Target JSON document as **String**.

Related Pages:

- jsonToClass() Operation
- Hiding Attributes From Interfaces
- JSON Type Mapping
- classToExtendedJSON() Operation
- extendedJSONToClass() Operation

`options`

The optional parameter is an object of type **ComposerOptions**.



Its attribute is:

Attribute	Type	Description	Allowed Values	
compact	Boolean	When **compact** is true, the JSON composer will generate compact JSON, otherwise it will generate pretty JSON. The default behavior of `classToJson()` (i.e. when called without options) is equivalent to **compact**=false.	true	Generate compact JSON.
			false	Generate pretty JSON (default).
keepNulls	Boolean	When **keepNulls** is true, attributes of anObject with NULL values will be rendered to the JSON object, otherwise they will be left out completely (see also chapter NULL Values). The default behavior of `classToJson()` (i.e. when called without options) is equivalent to **keepNulls**=false.	true	Render NULL attributes to the JSON object.
			false	Leave out NULL attributes (default).
writeTypeDiscriminator	Boolean	Runtime 2021.6 Use **writeTypeDiscriminator** to suppress the generation of xUML type properties ("e2e:type") to the generated JSON. If this option is true, the Runtime will write the original xUML type to the generated JSON in form of `"e2e:type": "<name of the original xUML type>"` if the type being serialized does not match the expected metadata. This is necessary if you want to convert the generated JSON back to an xUML class using jsonToClass(). Runtime versions before 2021.6 will ignore the value.	**true**	Write xUML type discriminator (default).
			false	Do not write xUML type discriminator.

Examples

```
set addressAsJSONDocument = myAddress.classToJSON();
```

Example File (Builder project Add-ons/JSON):

<your example path>\Add-ons\JSON\uml\userManager.xml