

# jsonToClass

Syntax	<pre>set anObject = aString.jsonToClass() set anObject = jsonToClass(literal)</pre>	
Semantics	<p>The operation <code>jsonToClass()</code> takes an JSON string (<code>aString</code>) and tries to map it to an <code>anObject</code>. If this is not possible, an error is raised.</p> <p>By default, the following mapping rules apply:</p> <ul style="list-style-type: none"><li>• JSON attributes are mapped to class attributes having the same name.</li><li>• Attributes not defined in the target class are ignored.</li></ul> <p>For more mapping details refer to <a href="#">JSON Type Mapping</a>.</p>	
Substitutables	aString	A JSON document as <b>String</b> .
	anObject	Target object, can be any object.
	literal	String literal.
Examples	<pre>set myAddress = addressAsJSONDocument.jsonToClass();</pre> <p>The action script below creates an object of type <b>Address</b>. An output object flow state named <b>myAddress</b> of type <b>Address</b> needs to be defined in the activity diagram.</p> <pre>create myAddress; set myAddress = addressAsJSONDocument.jsonToClass();</pre> <p>The example below shows how to specify the xUML type at runtime, e.g. if using a derived class.</p> <pre>{   "e2e:type": "urn:json.e2e.ch.AddressWithEmail",   "name": "John Snow",   "company": "Winter &amp; Partners",   "mailaddress": "john.snow@winter.com" }</pre> <div><div><div>Address</div><div>+name : String +company: String</div></div><div>→</div><div><div>AddressWithEm Mail</div><div>+email : String</div></div></div> <p>The xUML type is declared by the attribute "e2e:type". The type has to be specified like <code>&lt;xml_namespace&gt;.&lt;classname&gt;</code></p>	

## Related Pages:

- [JSON Type Mapping](#)
- [classToJSON\(\) Operation](#)
- [extendedJSONToClass\(\) Operation](#)
- [classToExtendedJSON\(\) Operation](#)

## Related Documentation:

- [JSON Homepage](#)
- [RFC 4627](#)

For more information on JSON refer to the [JSON Homepage](#) or to [RFC 4627](#).