

## TYPO3.Flow - Feature # 31261

<b>Status:</b>	New	<b>Priority:</b>	Should have
<b>Author:</b>	Christopher Hlubek	<b>Category:</b>	Object
<b>Created:</b>	2011-10-25	<b>Assigned To:</b>	
<b>Updated:</b>	2011-10-25	<b>Due date:</b>	
<b>PHP Version:</b>			
<b>Has patch:</b>	No		
<b>Complexity:</b>			
<b>Subject:</b>	Virtual objects - generate proxy classes for interfaces		
<b>Description</b>			
<p>It should be possible to declare an interface and configure an object (e.g. singleton based on a prototype with arguments) for it in Objects.yaml. Right now you have to write a factory class to have different singletons based on the same prototype object.</p>			
<b>Example:</b>			
<b>Objects.yaml</b>			
<pre>TYPO3\Example\MyInterface:   scope: singleton   className: TYPO3\Example\MyClass # &lt;- scope: prototype   arguments:     1:       value: Foo  TYPO3\Example\MyDebugInterface:   scope: singleton   className: TYPO3\Example\MyClass # &lt;- scope: prototype   arguments:     1:       value: Bar   properties:     debug:       value: true</pre>			
<p>There should be a generated proxy class for the interface that implements the interface. Dependency injection with the interface type should work with that automatically.</p>			
<p>I think instantiating the interface with <code>new \TYPO3\Example\MyInterface()</code> has not to work since it's also contra-intuitive to the default PHP behaviour. So DI or the ObjectManager must be used for that.</p>			

### History

---