JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/NLGModule.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/simplenlg/framework/NLGFactory.html)
* [Next Class](http://docs.google.com/simplenlg/framework/PhraseCategory.html)
* [Frames](http://docs.google.com/index.html?simplenlg/framework/NLGModule.html)
* [No Frames](http://docs.google.com/NLGModule.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.3dy6vkm)|
* [Method](#id.4d34og8)

simplenlg.framework

## Class NLGModule

* java.lang.Object
  + simplenlg.framework.NLGModule

Direct Known Subclasses: [Aggregator](http://docs.google.com/simplenlg/aggregation/Aggregator.html), [HTMLFormatter](http://docs.google.com/simplenlg/format/english/HTMLFormatter.html), [MorphologyProcessor](http://docs.google.com/simplenlg/morphology/english/MorphologyProcessor.html), [NewAggregator](http://docs.google.com/simplenlg/aggregation/NewAggregator.html), [OrthographyProcessor](http://docs.google.com/simplenlg/orthography/english/OrthographyProcessor.html), [Realiser](http://docs.google.com/simplenlg/realiser/english/Realiser.html), [SyntaxProcessor](http://docs.google.com/simplenlg/syntax/english/SyntaxProcessor.html), [TextFormatter](http://docs.google.com/simplenlg/format/english/TextFormatter.html)

* public abstract class NLGModule  
  extends java.lang.Object  
  NLGModule is the base class that all processing modules extend from. This is abstract and cannot therefore be instantiated itself. The supplied processing modules that come pre-packaged in SimpleNLG can be found within their appropriate packages.  
  All processing modules perform realisation on a tree of NLGElements. The modules can alter the tree in whichever way they wish. For example, the syntax processor replaces phrase elements with list elements consisting of inflected words while the morphology processor replaces inflected words with string elements.  
  **N.B.** the use of *module*, *processing module* and *processor* is interchangeable. They all mean an instance of this class.Version: 4.0 Author: D. Westwater, University of Aberdeen.

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [**NLGModule**](http://docs.google.com/simplenlg/framework/NLGModule.html#NLGModule())() |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + [Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) | * + [**getLexicon**](http://docs.google.com/simplenlg/framework/NLGModule.html#getLexicon())() Retrieves the lexicon currently being used by this module. |
| * + abstract void | * + [**initialise**](http://docs.google.com/simplenlg/framework/NLGModule.html#initialise())() Performs one-time initialisation of the module. |
| * + abstract java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> | * + [**realise**](http://docs.google.com/simplenlg/framework/NLGModule.html#realise(java.util.List))(java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> elements) Realises a List of NLGElements usually by iteratively calling the realise(NLGElement) method on each element in the list and adding the result into a new a List |
| * + abstract [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) | * + [**realise**](http://docs.google.com/simplenlg/framework/NLGModule.html#realise(simplenlg.framework.NLGElement))([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) element) Realises the given element. |
| * + void | * + [**setLexicon**](http://docs.google.com/simplenlg/framework/NLGModule.html#setLexicon(simplenlg.lexicon.Lexicon))([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) newLexicon) Sets the lexicon to be used by this module. |

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### NLGModule public NLGModule()

### Method Detail

#### initialise public abstract void initialise() Performs one-time initialisation of the module.

#### realise public abstract [NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) realise([NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html) element) Realises the given element. This call is usually recursive as the call processes the child elements of the given element.Parameters:element - the NLGElement to be realised. Returns:the NLGElement representing the realised state. This may be the initial element in a changed form or be a completely new element.

#### realise public abstract java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> realise(java.util.List<[NLGElement](http://docs.google.com/simplenlg/framework/NLGElement.html)> elements) Realises a List of NLGElements usually by iteratively calling the realise(NLGElement) method on each element in the list and adding the result into a new a ListParameters:elements - the List of NLGElements to be realised. Returns:the List of realised NLGElements.

#### setLexicon public void setLexicon([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) newLexicon) Sets the lexicon to be used by this module. Passing in null will remove the existing lexicon and no lexicon will be used.Parameters:newLexicon - the new Lexicon to be used.

#### getLexicon public [Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) getLexicon() Retrieves the lexicon currently being used by this module.Returns:the Lexicon in use. This will be null if there is currently no lexicon associated with this module.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/NLGModule.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/simplenlg/framework/NLGFactory.html)
* [Next Class](http://docs.google.com/simplenlg/framework/PhraseCategory.html)
* [Frames](http://docs.google.com/index.html?simplenlg/framework/NLGModule.html)
* [No Frames](http://docs.google.com/NLGModule.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.3dy6vkm)|
* [Method](#id.4d34og8)