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/MultipleLexicon.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/lexicon/LexiconInterface.html)
* [Next Class](http://docs.google.com/simplenlg/lexicon/NIHDBLexicon.html)
* [Frames](http://docs.google.com/index.html?simplenlg/lexicon/MultipleLexicon.html)
* [No Frames](http://docs.google.com/MultipleLexicon.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.1t3h5sf)|
* [Method](#id.17dp8vu)

simplenlg.lexicon

## Class MultipleLexicon

* java.lang.Object
  + [simplenlg.lexicon.Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html)
    - simplenlg.lexicon.MultipleLexicon
* public class MultipleLexicon  
  extends [Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html)  
  This class contains a set of lexicons, which are searched in order for the specified wordAuthor: ereiter

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [**MultipleLexicon**](http://docs.google.com/simplenlg/lexicon/MultipleLexicon.html#MultipleLexicon())() create an empty multi lexicon |
| * + [**MultipleLexicon**](http://docs.google.com/simplenlg/lexicon/MultipleLexicon.html#MultipleLexicon(simplenlg.lexicon.Lexicon...))([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html)... lexicons) create a multi lexicon with the specified lexicons |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + void | * + [**addFinalLexicon**](http://docs.google.com/simplenlg/lexicon/MultipleLexicon.html#addFinalLexicon(simplenlg.lexicon.Lexicon))([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) lex) add lexicon at end of list (is searched last) |
| * + void | * + [**addInitialLexicon**](http://docs.google.com/simplenlg/lexicon/MultipleLexicon.html#addInitialLexicon(simplenlg.lexicon.Lexicon))([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) lex) add lexicon at beginning of list (is searched first) |
| * + void | * + [**close**](http://docs.google.com/simplenlg/lexicon/MultipleLexicon.html#close())() close the lexicon (if necessary) if lexicon does not need to be closed, this does nothing |
| * + java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> | * + [**getWords**](http://docs.google.com/simplenlg/lexicon/MultipleLexicon.html#getWords(java.lang.String,%20simplenlg.framework.LexicalCategory))(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) returns all Words which have the specified base form and category |
| * + java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> | * + [**getWordsByID**](http://docs.google.com/simplenlg/lexicon/MultipleLexicon.html#getWordsByID(java.lang.String))(java.lang.String id) returns a List of WordElement which have this ID. |
| * + java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> | * + [**getWordsFromVariant**](http://docs.google.com/simplenlg/lexicon/MultipleLexicon.html#getWordsFromVariant(java.lang.String,%20simplenlg.framework.LexicalCategory))(java.lang.String variant, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) returns Words which have an inflected form and/or spelling variant that matches the specified variant, and are in the specified category. |
| * + boolean | * + [**isAlwaysSearchAll**](http://docs.google.com/simplenlg/lexicon/MultipleLexicon.html#isAlwaysSearchAll())() |
| * + void | * + [**setAlwaysSearchAll**](http://docs.google.com/simplenlg/lexicon/MultipleLexicon.html#setAlwaysSearchAll(boolean))(boolean alwaysSearchAll) |

### Methods inherited from class simplenlg.lexicon.[**Lexicon**](http://docs.google.com/simplenlg/lexicon/Lexicon.html)[getDefaultLexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getDefaultLexicon()), [getWord](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWord(java.lang.String)), [getWord](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWord(java.lang.String,%20simplenlg.framework.LexicalCategory)), [getWordByID](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordByID(java.lang.String)), [getWordFromVariant](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordFromVariant(java.lang.String)), [getWordFromVariant](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordFromVariant(java.lang.String,%20simplenlg.framework.LexicalCategory)), [getWords](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWords(java.lang.String)), [getWordsFromVariant](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordsFromVariant(java.lang.String)), [hasWord](http://docs.google.com/simplenlg/lexicon/Lexicon.html#hasWord(java.lang.String)), [hasWord](http://docs.google.com/simplenlg/lexicon/Lexicon.html#hasWord(java.lang.String,%20simplenlg.framework.LexicalCategory)), [hasWordByID](http://docs.google.com/simplenlg/lexicon/Lexicon.html#hasWordByID(java.lang.String)), [hasWordFromVariant](http://docs.google.com/simplenlg/lexicon/Lexicon.html#hasWordFromVariant(java.lang.String)), [hasWordFromVariant](http://docs.google.com/simplenlg/lexicon/Lexicon.html#hasWordFromVariant(java.lang.String,%20simplenlg.framework.LexicalCategory)), [lookupWord](http://docs.google.com/simplenlg/lexicon/Lexicon.html#lookupWord(java.lang.String)), [lookupWord](http://docs.google.com/simplenlg/lexicon/Lexicon.html#lookupWord(java.lang.String,%20simplenlg.framework.LexicalCategory))

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

### Constructor Detail

#### MultipleLexicon public MultipleLexicon() create an empty multi lexicon

#### MultipleLexicon public MultipleLexicon([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html)... lexicons) create a multi lexicon with the specified lexiconsParameters:lexicons -

### Method Detail

#### addInitialLexicon public void addInitialLexicon([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) lex) add lexicon at beginning of list (is searched first)Parameters:lex -

#### addFinalLexicon public void addFinalLexicon([Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) lex) add lexicon at end of list (is searched last)Parameters:lex -

#### isAlwaysSearchAll public boolean isAlwaysSearchAll()Returns:the alwaysSearchAll

#### setAlwaysSearchAll public void setAlwaysSearchAll(boolean alwaysSearchAll)Parameters:alwaysSearchAll - the alwaysSearchAll to set

#### getWords public java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> getWords(java.lang.String baseForm, [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category) **Description copied from class:**[**Lexicon**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWords(java.lang.String,%20simplenlg.framework.LexicalCategory)) returns all Words which have the specified base form and category**Specified by:** [getWords](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWords(java.lang.String,%20simplenlg.framework.LexicalCategory)) in class [Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) Parameters:baseForm - - base form of word, eg "be" or "dog" (not "is" or "dogs")category - - syntactic category of word (ANY for unknown) Returns:collection of all matching Words (may be empty)

#### getWordsByID public java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> getWordsByID(java.lang.String id) **Description copied from class:**[**Lexicon**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordsByID(java.lang.String)) returns a List of WordElement which have this ID. IDs are lexicon-dependent, and should be unique. Therefore the list should contain either zero elements (if no such word exists) or one element (if the word is found)**Specified by:** [getWordsByID](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordsByID(java.lang.String)) in class [Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) Parameters:id - - internal lexicon ID for a word Returns:either empty list (if no word with this ID exists) or list containing the matching word

* + - getWordsFromVariant  
      public java.util.List<[WordElement](http://docs.google.com/simplenlg/framework/WordElement.html)> getWordsFromVariant(java.lang.String variant,  
       [LexicalCategory](http://docs.google.com/simplenlg/framework/LexicalCategory.html) category)  
      **Description copied from class:**[**Lexicon**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordsFromVariant(java.lang.String,%20simplenlg.framework.LexicalCategory))  
      returns Words which have an inflected form and/or spelling variant that matches the specified variant, and are in the specified category.  
      *Note:* the returned word list may not be complete, it depends on how it is implemented by the underlying lexicon**Specified by:** [getWordsFromVariant](http://docs.google.com/simplenlg/lexicon/Lexicon.html#getWordsFromVariant(java.lang.String,%20simplenlg.framework.LexicalCategory)) in class [Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html) Parameters:variant - - base form, inflected form, or spelling variant of wordcategory - - syntactic category of word (ANY for unknown) Returns:list of all matching Words (empty list if no matching WordElement found)

#### close public void close() **Description copied from class:**[**Lexicon**](http://docs.google.com/simplenlg/lexicon/Lexicon.html#close()) close the lexicon (if necessary) if lexicon does not need to be closed, this does nothing**Overrides:** [close](http://docs.google.com/simplenlg/lexicon/Lexicon.html#close()) in class [Lexicon](http://docs.google.com/simplenlg/lexicon/Lexicon.html)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/MultipleLexicon.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/lexicon/LexiconInterface.html)
* [Next Class](http://docs.google.com/simplenlg/lexicon/NIHDBLexicon.html)
* [Frames](http://docs.google.com/index.html?simplenlg/lexicon/MultipleLexicon.html)
* [No Frames](http://docs.google.com/MultipleLexicon.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#id.3znysh7)|
* [Method](#id.2et92p0)
* Detail:
* Field |
* [Constr](#id.1t3h5sf)|
* [Method](#id.17dp8vu)