EWN SERVICE

getFOCoordinates

```
public java.util.List<ResultIli>
getFOCoordinates(java.lang.String key,
    java.lang.String partOfSpeech,
    java.lang.String lang,
    java.lang.String word)
```

get the coordinates of a given word, i.e. its siblings (words sharing the same ancestor with the given word, including the word itself and its synonyms); on demand, hypernyms are being added to the result

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`word` - the word for which the coordinates are looked for

Returns:

a list of mappings between word readings and their siblings

---

getFOCoordinates_includeHypernyms

```
public java.util.List<ResultIli>
getFOCoordinates_includeHypernyms(java.lang.String key,
     java.lang.String partOfSpeech,
     java.lang.String lang,
     java.lang.String word,
     boolean includeHypernyms)
```

get the coordinates of a given word, i.e. its siblings (words sharing the same ancestor with the given word, including the word itself and its synonyms); on demand, hypernyms are being added to the result

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`word` - the word for which the coordinates are looked for

`includeHypernyms` - specify whether word's hypernyms are to be added to the result

Returns:

a list of mappings between word readings and their siblings

---

getFOElements

```
public java.util.List<ResultIli> getFOElements(java.lang.String key,
   java.lang.String partOfSpeech,
   java.lang.String lang,
   java.lang.String word)
```

for a given word get its siblings, its hypernyms and its hyponyms

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`word` - the word for which the coordinates are looked for

Returns:

a list of mappings between word ilis and their first order elements

---

getFOHypernymILIs

```
public java.util.List<ResultIli>
getFOHypernymILIs(java.lang.String key,
    java.lang.String partOfSpeech,
    java.lang.String lang,
    java.lang.String word)
```

look for the hypernym ilis of a given word. REMARK: a word reading (synset) can have more than one hypernym; this method creates a mapping between ili and hypernym ilis

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`word` - the word for which the coordinates are looked for

Returns:

a list of mappings between ilis and their corresponding hypernym ilis

---

getFOHypernyms_selfIncluded

```
public java.util.List<ResultIli>
getFOHypernyms_selfIncluded(java.lang.String key,
    java.lang.String partOfSpeech,
    java.lang.String lang,
    java.lang.String word,
    boolean selfIncluded)
```

look for the hypernyms (myWord IS-A X) of a given word. REMARK: a word reading (synset) can have more than one hypernym; this method puts them alltogether mapped to the same ili

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`word` - he word for which the hypernyms are looked for

`selfIncluded` - specify whether the word's synset should be added to the return value

Returns:

a list of mappings between ili numbers corresponding to each word reading and the hypernyms of it (the synset itself is being added when "selfIncluded" is true)

---

getFOHypernyms

```
public java.util.List<ResultIli> getFOHypernyms(java.lang.String key,
   java.lang.String partOfSpeech,
   java.lang.String lang,
   java.lang.String word)
```

look for the hypernyms (myWord IS-A X) of a given word. REMARK: a word reading (synset) can have more than one hypernym; this method puts them alltogether mapped to the same ili

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`word` - he word for which the hypernyms are looked for

`specify` - whether the word's synset should be added to the return value

Returns:

a list of mappings between ili numbers corresponding to each word reading and the hypernyms of it

---

getFOHypernymsByILI

```
public java.util.List<ResultIli>
getFOHypernymsByILI(java.lang.String key,
    java.lang.String partOfSpeech,
    java.lang.String lang,
    java.lang.String ili)
```

Look for the hypernyms of a given word reading/ili (ILI). REMARK: a word reading (synset) can have more than one hypernym; this method creates a mapping between hypernym ili and hypernym lexical representation

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query
`ili` - the synsets Interlingua Link ID (ILI) which the hypernyms are to be looked for. You have to make sure that it consists only of digits!
Returns:
a list of mappings between hypernym ilis and their lexical representation

---

getFOHyponymILIs

```
public java.util.List<ResultIli>
getFOHyponymILIs(java.lang.String key,
    java.lang.String partOfSpeech,
    java.lang.String lang,
    java.lang.String word)
```

look for the hyponym ilis of a given word. REMARK: a word reading (synset) can have more than one hyponym; this method creates a mapping between ili and hyponym ilis
Parameters:
`key` - user key for using service
`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective
`lang` - the language of the query
`word` - he word for which the hyponym ilis are looked for
Returns:
a list of mappings between ilis and their corresponding hyponym ilis of each reading/sense

---

getFOHyponyms

```
public java.util.List<ResultIli> getFOHyponyms(java.lang.String key,
   java.lang.String partOfSpeech,
   java.lang.String lang,
   java.lang.String word)
```

look for the hyponyms (X IS-A myWord) of a given word. REMARK: a word reading (synset) can have more than one hyponym; this method puts them alltogether mapped to the same ili
Parameters:
`key` - user key for using service
`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective
`lang` - the language of the query
`word` - the word for which the hyponyms are looked for
Returns:
a list of mappings between ili numbers corresponding to each word reading and the hyponyms of it

---

getFOHyponyms_selfIncluded

```
public java.util.List<ResultIli>
getFOHyponyms_selfIncluded(java.lang.String key,
    java.lang.String partOfSpeech,
    java.lang.String lang,
    java.lang.String word,
    boolean selfIncluded)
```

look for the hyponyms (X IS-A myWord) of a given word. REMARK: a word reading (synset) can have more than one hyponym; this method puts them alltogether mapped to the same ili

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`word` - the word for which the hyponyms are looked for

`selfIncluded` - pecify whether the word's synset should be added to the return value

Returns:

a list of mappings between ili numbers corresponding to each word reading and the hyponyms of it (the synset itself is being added when "selfIncluded" is true)

---

getFOHyponymsByILI

```
public java.util.List<ResultIli>
getFOHyponymsByILI(java.lang.String key,
    java.lang.String partOfSpeech,
    java.lang.String lang,
    java.lang.String ili)
```

look for the hyponyms of a given word reading/ili. REMARK: a word reading (synset) can have more than one hyponym; this method creates a mapping between hyponym ili and hyponym lexical representation

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`ili` - the ili marking the intended reading for which the hyponyms are to be looked for

Returns:

a list of mappings between hyponym ilis and their lexical representation

---

getFORelatives

```
public java.util.List<ResultIli> getFORelatives(java.lang.String key,
```

```
   java.lang.String partOfSpeech,
   java.lang.String lang,
   java.lang.String word)
```

for a given word get its hypernyms and its hyponyms

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`word` - the word for which the coordinates are looked for

Returns:

---

getFORelatives_selfIncluded

```
public java.util.List<ResultIli>
getFORelatives_selfIncluded(java.lang.String key,
    java.lang.String partOfSpeech,
    java.lang.String lang,
    java.lang.String word,
    boolean selfIncluded)
```

for a given word get its hypernyms and its hyponyms

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`word` - the word for which the coordinates are looked for

`selfIncluded` - specify whether the word's synset should be added to the return value

Returns:

a list of mappings between word readings/ilis and their relatives (hypernyms + hyponyms)

---

getGlossByILI

```
public java.lang.String getGlossByILI(java.lang.String key,
   java.lang.String partOfSpeech,
   java.lang.String lang,
   java.lang.String ili)
```

Get definition for a given synset

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`ili` - the synset's ILI - this String should only contain digits!

Returns:

a String containing the gloss, an empty string if there is no gloss in the database

---

getGlosses

```
public java.util.List<ResultIli> getGlosses(java.lang.String key,
   java.lang.String partOfSpeech,
   java.lang.String lang,
   java.lang.String word)
```

Get all synset definition for a word

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`word` - the word to be looked for

Returns:

a set of mappings between ILIs (for the different word meanings) * and definitions

---

getILIs

```
public java.util.List<java.lang.String> getILIs(java.lang.String key,
   java.lang.String partOfSpeech,
   java.lang.String lang,
   java.lang.String word)
```

Look up the Interlingua IDs (ILIs, ilis) of a word.

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`word` - the word to be looked for

Returns:

a set of ILIs that represent meanings of the given word

---

getLexemesByILISet

```
public java.util.List<java.lang.String>
getLexemesByILISet(java.lang.String key,
    java.lang.String partOfSpeech,
    java.lang.String lang,
    java.util.List<java.lang.String> ilis)
```

get lexical representations for a list of given ilis

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective
`lang` - the language of the query
`ilis` - a list of ilis

Returns:
a list of lexical representations

---

getLexemesByILI

```
public java.util.List<java.lang.String>
getLexemesByILI(java.lang.String key,
    java.lang.String partOfSpeech,
    java.lang.String lang,
    java.lang.String ili)
```

Get lexical representation of a synset specified by its inter-lingual index.

Parameters:
`key` - user key for using service
`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective
`lang` - the language of the query
`ili` - the inter-lingual index

Returns:
a set of Strings of all words/entries that represent the given ILI

---

getRelation

```
public java.util.List<ResultIli> getRelation(java.lang.String key,
   java.lang.String partOfSpeech,
   java.lang.String lang,
   java.lang.String word,
   java.lang.String relation)
```

Queries an relation in the wordnet. Read the EuroWordNet manual in order to find out what relations exist.

Parameters:
`key` - user key for using service
`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective
`lang` - the language of the query
`word` - a word
`relation` - the EWN name of the relation

Returns:
a map with the word's different ILIs as keys and sets of the relation targets as values.

---

getRelationByILI

```
public java.util.List<java.lang.String>
getRelationByILI(java.lang.String key,
    java.lang.String partOfSpeech,
    java.lang.String lang,
    java.lang.String ili,
    java.lang.String relation)
```

Queries any relation in the wordnet. When using this query, make sure you have set the correct part of speech for the results that you expect. If you are not sure, it is better to call setPartOfSpeech(POS_ANY) before. Read the EuroWordNet manual in order

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`ili` - the concept ILI

`relation` - the EuroWordNet name of the relation

Returns:

a set of ILIs where the relation points to from the given ili

---

getRelationByILI_ewnRelation

```
public java.util.List<java.lang.String>
getRelationByILI_ewnRelation(java.lang.String key,
     java.lang.String partOfSpeech,
     java.lang.String lang,
     java.lang.String ili,
     java.lang.String relation,
     boolean ewnRelation)
```

Queries any relation in the wordnet. When using this query, make sure you have set the correct part of speech for the results that you expect. If you are not sure, it is better to call setPartOfSpeech(POS_ANY) before. Read the EuroWordNet manual in order

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`ili` - the concept ili

`relation` - the name of the relation, either the EWN or the WordNet name

`ewnRelation` - determines whether an EWN (true) or a WordNet (false) relation name is used.

Returns:

a set of ILIs where the relation points to from the given ili

---

getSenses

```
public java.util.List<ResultIli> getSenses(java.lang.String key,
   java.lang.String partOfSpeech,
   java.lang.String lang,
   java.lang.String word)
```

look up a word in the lexical resource (EWN)

Parameters:

`key` - user key for using service

`partOfSpeech` - a value from the following set {n, v, a} corresponding to noun, verb, adjective

`lang` - the language of the query

`word` - the word to be looked for

Returns:

a list of mappings between ili numbers and synonyms pertaining to each reading of the input word

ResultIli

getIli

```
public java.lang.String getIli()
```

gets a synsets Interlingua Link ID (ILI)

Returns:

a Interlingua Link ID (ILI)

---

setIli

```
public void setIli(java.lang.String ili)
```

set a synsets Interlingua Link ID (ILI)

Parameters:

`Interlingua` - Link ID (ILI)

---

getTerms

```
public java.lang.String[] getTerms()
```

gets terms related to an ILI

Returns:

terms related to an ILI

---

setTerms

```
public void setTerms(java.lang.String[] terms)
```

set a list of terms
Parameters:
`list` - of terms