ToOWLConverterApi

Generated by Doxygen 1.8.12

Contents

1 Hierarchical Index 1

1.1 Class Hierarchy 1

2 Class Index 3

2.1 Class List 3

3 Class Documentation 5

3.1 de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory Class Reference 5

3.1.1 Detailed Description 5

3.1.2 Constructor & Destructor Documentation 6

3.1.2.1 AbstractOWLOntologyFactory() [1/2] 6

3.1.2.2 AbstractOWLOntologyFactory() [2/2] 6

3.1.3 Member Function Documentation 6

3.1.3.1 addConceptRecursivly() 6

3.1.3.2 addProperties() 7

3.1.3.3 addProperty() 7

3.1.3.4 addRelation() 7

3.1.3.5 addRelations() 8

3.1.3.6 loadOntology() 8

3.1.3.7 saveOntology() 8

3.2 de.dailab.nsm.decomposition.owlConverter.model.OWLConcept Interface Reference 9

3.2.1 Detailed Description 9

3.3 de.dailab.nsm.decomposition.owlConverter.model.OWLNamedEntity Interface Reference 9

3.3.1 Detailed Description 10

3.4 de.dailab.nsm.decomposition.owlConverter.util.OWLNamespace Enum Reference 10

3.4.1 Detailed Description 10

3.5 de.dailab.nsm.decomposition.owlConverter.OWLOntologyFactoryTest Class Reference 11

3.5.1 Detailed Description 11

3.6 de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider Class Reference . . . 11

3.6.1 Detailed Description 12

3.7 de.dailab.nsm.decomposition.owlConverter.model.OWLProperty Interface Reference 12

3.7.1 Detailed Description 12

3.8 de.dailab.nsm.decomposition.owlConverter.model.OWLRelation Interface Reference 13

3.8.1 Detailed Description 13

3.9 de.dailab.nsm.decomposition.owlConverter.model.OWLRelationType Interface Reference 13

3.9.1 Detailed Description 13

3.10 de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider.OWLTestConcept Class Reference 14

3.11 de.dailab.nsm.decomposition.owlConverter.Parser Interface Reference 14

3.11.1 Detailed Description 14

Index **15**

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory 5
- de.dailab.nsm.decomposition.owlConverter.model.OWLNamedEntity 9
  - de.dailab.nsm.decomposition.owlConverter.model.OWLConcept 9
    - de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider.OWLTestConcept . . . 14
  - de.dailab.nsm.decomposition.owlConverter.model.OWLRelation 13
  - de.dailab.nsm.decomposition.owlConverter.model.OWLRelationType 13
- de.dailab.nsm.decomposition.owlConverter.util.OWLNamespace 10
- de.dailab.nsm.decomposition.owlConverter.OWLOntologyFactoryTest 11
- de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider 11
- de.dailab.nsm.decomposition.owlConverter.model.OWLProperty 12
- de.dailab.nsm.decomposition.owlConverter.Parser 14

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory 5
de.dailab.nsm.decomposition.owlConverter.model.OWLConcept 9
de.dailab.nsm.decomposition.owlConverter.model.OWLNamedEntity 9
de.dailab.nsm.decomposition.owlConverter.util.OWLNamespace 10
de.dailab.nsm.decomposition.owlConverter.OWLOntologyFactoryTest 11
de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider 11
de.dailab.nsm.decomposition.owlConverter.model.OWLProperty 12
de.dailab.nsm.decomposition.owlConverter.model.OWLRelation 13
de.dailab.nsm.decomposition.owlConverter.model.OWLRelationType 13
de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider.OWLTestConcept 14
de.dailab.nsm.decomposition.owlConverter.Parser 14

Chapter 3

Class Documentation

3.1 de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory Class Reference

Public Member Functions

- AbstractOWLOntologyFactory () throws OWLOntologyCreationException
- AbstractOWLOntologyFactory (OWLOntology ontology) throws OWLOntologyCreationException
- void saveOntology (IRI documentIRI) throws OWLOntologyStorageException
- OWLNamedIndividual addConceptRecursivly (OWLConcept concept)
- void addProperties (Collection< OWLProperty > properties, OWLNamedIndividual owlIndividual)
- void addRelations (Collection< OWLRelation > relations)

Static Public Member Functions

- static OWLOntology loadOntology (IRI ontologyDocument) throws OWLOntologyCreationException

Protected Member Functions

- abstract OWLAxiom addProperty (OWLProperty property, OWLIndividual owlIndividual)
- abstract OWLAxiom addRelation (OWLRelation relation)

Protected Attributes

- final OWLDataFactory **factory**

3.1.1 Detailed Description

This factory wraps the OWLDataFactory. By constructing this factory, an owlManager, an owlFactory and an owl↩ Ontology will be created. An owlManager manages only one ontology at a time. So this factory manages only one ontology at a time too. You can create a new ontology or work with a given one to extend it with basic owl components.

All changes of the ontology are collected in 'changes' and applied at once for a better performance.

Feel free to extend the factory for further components.

Author

Maik

3.1.2 Constructor & Destructor Documentation

3.1.2.1 AbstractOWLOntologyFactory() [1/2]

```
de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory.AbstractOWLOntologyFactory
( ) throws OWLOntologyCreationException
```

Creates a new factory and a new ontology, the factory works with.

Exceptions

OWLOntologyCreationException	Occurs when an ontology with the new generated ID already exists, thus never our fault.

3.1.2.2 AbstractOWLOntologyFactory() [2/2]

```
de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory.AbstractOWLOntologyFactory
(
             OWLOntology ontology ) throws OWLOntologyCreationException
```

Creates a new factory for a given ontology.

Parameters

ontology	Ontology the factory works with.

Exceptions

OWLOntologyCreationException	Occurs when an ontology with the given IRI already exists.

3.1.3 Member Function Documentation

3.1.3.1 addConceptRecursivly()

```
OWLNamedIndividual de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory.add↩
ConceptRecursivly (
             OWLConcept concept )
```

Adds a whole concept with all it's parts to the ontology. Contains check of every entity is done by the OWLDataFactory.

Precondition

All constraints of the semantic network hold so that a further check is not needed.
Ontology and concept are not null.

Postcondition

Concept, all it's relations and target concepts will add recursivly to the ontology as an axiom.

Parameters

ontology	The ontology the concept shall be add to. (not null)
concept	The cocept to add to the ontology. (not null)

3.1.3.2 addProperties()

```
void de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory.addProperties (
            Collection< OWLProperty > properties,
            OWLNamedIndividual owlIndividual )
```

Adds properties to an OWLIndividual.

Parameters

properties	Properties to add to the individual. (not null)
owlIndividual	The individual the properties shall add to. It has to be part of the ontology signature. (not null)

3.1.3.3 addProperty()

```
abstract OWLAxiom de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory.add↵
Property (
            OWLProperty property,
            OWLIndividual owlIndividual )  [abstract], [protected]
```

Adds a property to the specified individual of the ontology. Each ToOWLConverter shall decide itself how to add a property internally, so it's possible to use owl2 features like negation, symmetric, reflexive, ... properties

Parameters

property	
owlIndividual	

Returns

3.1.3.4 addRelation()

```
abstract OWLAxiom de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory.add↵
Relation (
            OWLRelation relation )  [abstract], [protected]
```

Adds a relation to the ontology. Each ToOWLConverter shall decide itself how to add a relation internally.

Parameters

relation	Relation to add to the ontology

Returns

OWLAxiom which describes the addition of the relation

3.1.3.5 addRelations()

```
void de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory.addRelations (
             Collection< OWLRelation > relations )
```

Adds relations to an OWLIndividual.

Parameters

relations	Relations to add to the individual. (not null)
owlIndividual	The individual the relations shall add to. (not null)

3.1.3.6 loadOntology()

```
static OWLOntology de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory.load↩
Ontology (
             IRI ontologyDocument ) throws OWLOntologyCreationException  [static]
```

Loads an OWLOntology from the given document IRI.

Parameters

ontologyDocument	IRI specifier of the owl document file.

Returns

Loaded OWLOntology.

Exceptions

OWLOntologyCreationException	Occurs when something went wrong. Thrown by OWLOntologyManager.

3.1.3.7 saveOntology()

```
void de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory.saveOntology (
             IRI documentIRI ) throws OWLOntologyStorageException
```

Saves the managed ontology to the specified document.

Parameters

documentIRI	IRI specifier of the owl document file.

Exceptions

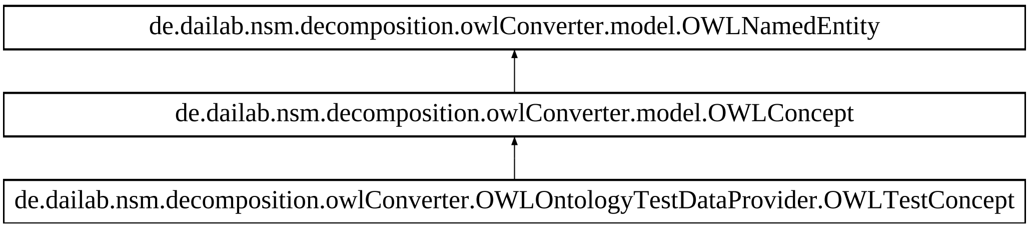
OWLOntologyStorageException	Occurs when something went wrong. Thrown by OWLOntologyManager.

The documentation for this class was generated from the following file:

- src/main/java/de/dailab/nsm/decomposition/owlConverter/AbstractOWLOntologyFactory.java

3.2 de.dailab.nsm.decomposition.owlConverter.model.OWLConcept Interface Reference

Inheritance diagram for de.dailab.nsm.decomposition.owlConverter.model.OWLConcept:


de.dailab.nsm.decomposition.owlConverter.model.OWLNamedEntity
de.dailab.nsm.decomposition.owlConverter.model.OWLConcept
de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider.OWLTestConcept

Public Member Functions

- String **getType** ()
- Collection< OWLRelation > **getRelations** ()
- void **addRelations** (Collection< OWLRelation > relations)
- Collection< OWLProperty > **getProperties** ()
- void **addProperties** (Collection< OWLProperty > properties)

3.2.1 Detailed Description

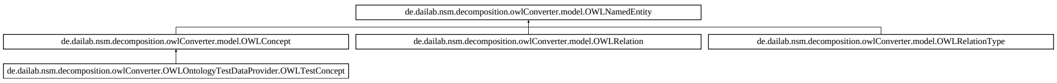
Represent common nodes in the OWL graph.

The documentation for this interface was generated from the following file:

- src/main/java/de/dailab/nsm/decomposition/owlConverter/model/OWLConcept.java

3.3 de.dailab.nsm.decomposition.owlConverter.model.OWLNamedEntity Interface Reference

Inheritance diagram for de.dailab.nsm.decomposition.owlConverter.model.OWLNamedEntity:


de.dailab.nsm.decomposition.owlConverter.model.OWLNamedEntity
de.dailab.nsm.decomposition.owlConverter.model.OWLConcept
de.dailab.nsm.decomposition.owlConverter.model.OWLRelation
de.dailab.nsm.decomposition.owlConverter.model.OWLRelationType
de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider.OWLTestConcept

Public Member Functions

- String **getName** ()

3.3.1 Detailed Description

OWLEntities represent the vertexes in the OWL graph.

The documentation for this interface was generated from the following file:

- src/main/java/de/dailab/nsm/decomposition/owlConverter/model/OWLNamedEntity.java

3.4 de.dailab.nsm.decomposition.owlConverter.util.OWLNamespace Enum Reference

Public Member Functions

- **OWLNamespace** (String name)
- String **toString** ()

Static Public Member Functions

- static String **removeIllegalNamespaceCharacters** (String entity)
- static String **removeNullNames** (String entity)

Public Attributes

- **OWL_NAMESPACE_PREFIX** =("http://www.dailab.de/ontologies/")
- **OWL_ONTOLOGY_NAMESPACE** =(OWL_NAMESPACE_PREFIX + "ontology#")
- **OWL_CLASS_NAMESPACE** =(OWL_NAMESPACE_PREFIX + "class#")
- **OWL_RELATION_NAMESPACE** =(OWL_NAMESPACE_PREFIX + "relation#")
- **OWL_INDIVIDUAL_NAMESPACE** =(OWL_NAMESPACE_PREFIX + "individual#")

3.4.1 Detailed Description

Provides common OWL namespaces and some internal conventions for this OWLConverter

The documentation for this enum was generated from the following file:

- src/main/java/de/dailab/nsm/decomposition/owlConverter/util/OWLNamespace.java

3.5 de.dailab.nsm.decomposition.owlConverter.OWLOntologyFactoryTest Class Reference

Public Member Functions

- void **init** () throws OWLOntologyCreationException
- void **testAddConcept** ()
- void **testAddConceptWithProperties** ()
- void **testAddConceptWithRelations** ()
- void **testAddCylicConcepts** ()

3.5.1 Detailed Description

Tests the AbstractOWLOntologyFactory functions and a the avoidance of cyles while creating concepts.

The documentation for this class was generated from the following file:

- src/test/java/de/dailab/nsm/decomposition/owlConverter/OWLOntologyFactoryTest.java

3.6 de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider Class Reference

Classes

- class OWLTestConcept

Protected Member Functions

- OWLConcept **createOWLConcept** (String name, String type)
- Collection< OWLRelation > **getTestRelations** (String entity1, String entity2, String entity3)

Static Protected Member Functions

- static AbstractOWLOntologyFactory **createTestFactory** () throws OWLOntologyCreationException
- static OWLProperty **createOWLDataProperty** (String valueAsString, OWL2Datatype type)
- static OWLRelation **createOWLRelation** (String name, OWLConcept source, OWLConcept target, OWLRelationType type)
- static OWLRelationType **createOWLRelationType** (String name)
- static Collection< OWLProperty > **getTestProperties** ()
- static OWLOntology **getTestOntology** (AbstractOWLOntologyFactory factory) throws IOException, OWLOntologyStorageException, OWLOntologyCreationException

Static Protected Attributes

- static final String **TEMP_DATA_PREFIX** = "temp"
- static final String **TEMP_DATA_SUFFIX** = ".owl"
- static final String **CONCEPT1** = "concept1"
- static final String **CONCEPT2** = "concept2"
- static final String **CONCEPT3** = "concept3"
- static final String **OBJECT_TYPE** = "object"
- static final String **FISH_TYPE** = "fish"
- static final String **TREE_TYPE** = "tree"
- static final String **ANIMAL_TYPE** = "animal"
- static final String **DEER_TYPE** = "deer"
- static final String **IS_NEXT_TO_RELATION_TYPE** = "is_next_to"
- static final String **SYNONYMY_TPYE** = "synonymy"
- static final String **ANTONYMY_TYPE** = "antonymy"
- static final String **HYPONYMY_TYPE** = "hyponymy"
- static final String **TEST_RELATION_1** = "testRelation1"
- static final String **TEST_RELATION_2** = "testRelation2"
- static final String **TEST_RELATION_3** = "testRelation3"

3.6.1 Detailed Description

Provides test data for the ToOWLConverterAPI tests.

The documentation for this class was generated from the following file:

- src/test/java/de/dailab/nsm/decomposition/owlConverter/OWLOntologyTestDataProvider.java

3.7 de.dailab.nsm.decomposition.owlConverter.model.OWLProperty Interface Reference

Public Member Functions

- String **getValueAsString** ()
- OWL2Datatype **getType** ()

3.7.1 Detailed Description

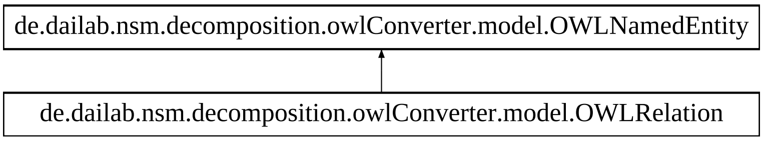
Interface for a property of an entity.

The documentation for this interface was generated from the following file:

- src/main/java/de/dailab/nsm/decomposition/owlConverter/model/OWLProperty.java

3.8 de.dailab.nsm.decomposition.owlConverter.model.OWLRelation Interface Reference

Inheritance diagram for de.dailab.nsm.decomposition.owlConverter.model.OWLRelation:


de.dailab.nsm.decomposition.owlConverter.model.OWLNamedEntity
de.dailab.nsm.decomposition.owlConverter.model.OWLRelation

Public Member Functions

- OWLConcept **getSource** ()
- OWLConcept **getTarget** ()
- OWLRelationType **getType** ()

3.8.1 Detailed Description

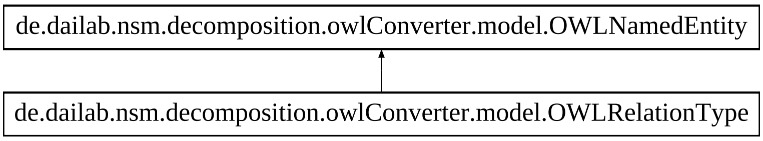
Interface for relations. A relation can be modelled as a vertex (to add properties for vertexes) or an edge in the graph.

The documentation for this interface was generated from the following file:

- src/main/java/de/dailab/nsm/decomposition/owlConverter/model/OWLRelation.java

3.9 de.dailab.nsm.decomposition.owlConverter.model.OWLRelationType Interface Reference

Inheritance diagram for de.dailab.nsm.decomposition.owlConverter.model.OWLRelationType:


de.dailab.nsm.decomposition.owlConverter.model.OWLNamedEntity
de.dailab.nsm.decomposition.owlConverter.model.OWLRelationType

Additional Inherited Members

3.9.1 Detailed Description

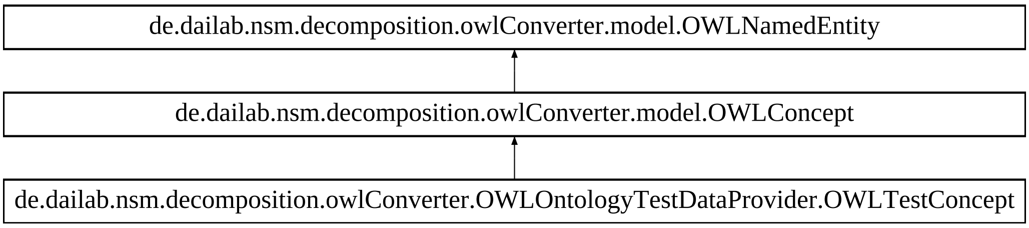
Interface for relation types.

The documentation for this interface was generated from the following file:

- src/main/java/de/dailab/nsm/decomposition/owlConverter/model/OWLRelationType.java

3.10 de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider.OWLTestConcept Class Reference

Inheritance diagram for de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider.OWLTestConcept:


de.dailab.nsm.decomposition.owlConverter.model.OWLNamedEntity
de.dailab.nsm.decomposition.owlConverter.model.OWLConcept
de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider.OWLTestConcept

Public Member Functions

- String **getName** ()
- Collection< OWLProperty > **getProperties** ()
- Collection< OWLRelation > **getRelations** ()
- String **getType** ()
- void **addProperties** (Collection< OWLProperty > properties)
- void **addRelations** (Collection< OWLRelation > relations)

Protected Member Functions

- **OWLTestConcept** (String name, String type, Collection< OWLProperty > properties, Collection< OWLRelation > relations)

The documentation for this class was generated from the following file:

- src/test/java/de/dailab/nsm/decomposition/owlConverter/OWLOntologyTestDataProvider.java

3.11 de.dailab.nsm.decomposition.owlConverter.Parser Interface Reference

Public Member Functions

- Collection< OWLConcept > **parse** (Graph< Concept, WeightedEdge > graph)

3.11.1 Detailed Description

Each ToOWLConverter needs a parser which parses the specific knowledge graph to an OWLOntology. Those parsers need to implement this interface.

The documentation for this interface was generated from the following file:

- src/main/java/de/dailab/nsm/decomposition/owlConverter/Parser.java

Index

- AbstractOWLOntologyFactory
  - de::dailab::nsm::decomposition::owlConverter::AbstractOWLOntologyFactory, 6
- addConceptRecursivly
  - de::dailab::nsm::decomposition::owlConverter::AbstractOWLOntologyFactory, 6
- addProperties
  - de::dailab::nsm::decomposition::owlConverter::AbstractOWLOntologyFactory, 7
- addProperty
  - de::dailab::nsm::decomposition::owlConverter::AbstractOWLOntologyFactory, 7
- addRelation
  - de::dailab::nsm::decomposition::owlConverter::AbstractOWLOntologyFactory, 7
- addRelations
  - de::dailab::nsm::decomposition::owlConverter::AbstractOWLOntologyFactory, 8

- de.dailab.nsm.decomposition.owlConverter.AbstractOWLOntologyFactory, 5
- de.dailab.nsm.decomposition.owlConverter.model.OWLConcept, 9
- de.dailab.nsm.decomposition.owlConverter.model.OWLNamedEntity, 9
- de.dailab.nsm.decomposition.owlConverter.model.OWLProperty, 12
- de.dailab.nsm.decomposition.owlConverter.model.OWLRelation, 13
- de.dailab.nsm.decomposition.owlConverter.model.OWLRelationType, 13
- de.dailab.nsm.decomposition.owlConverter.OWLOntologyFactoryTest, 11
- de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider, 11
- de.dailab.nsm.decomposition.owlConverter.OWLOntologyTestDataProvider.OWLTestConcept, 14
- de.dailab.nsm.decomposition.owlConverter.Parser, 14
- de.dailab.nsm.decomposition.owlConverter.util.OWLNamespace, 10
- de::dailab::nsm::decomposition::owlConverter::AbstractOWLOntologyFactory
  - AbstractOWLOntologyFactory, 6
  - addConceptRecursivly, 6
  - addProperties, 7
  - addProperty, 7
  - addRelation, 7
  - addRelations, 8
  - loadOntology, 8
  - saveOntology, 8

- loadOntology
  - de::dailab::nsm::decomposition::owlConverter::AbstractOWLOntologyFactory, 8

- saveOntology
  - de::dailab::nsm::decomposition::owlConverter::AbstractOWLOntologyFactory, 8