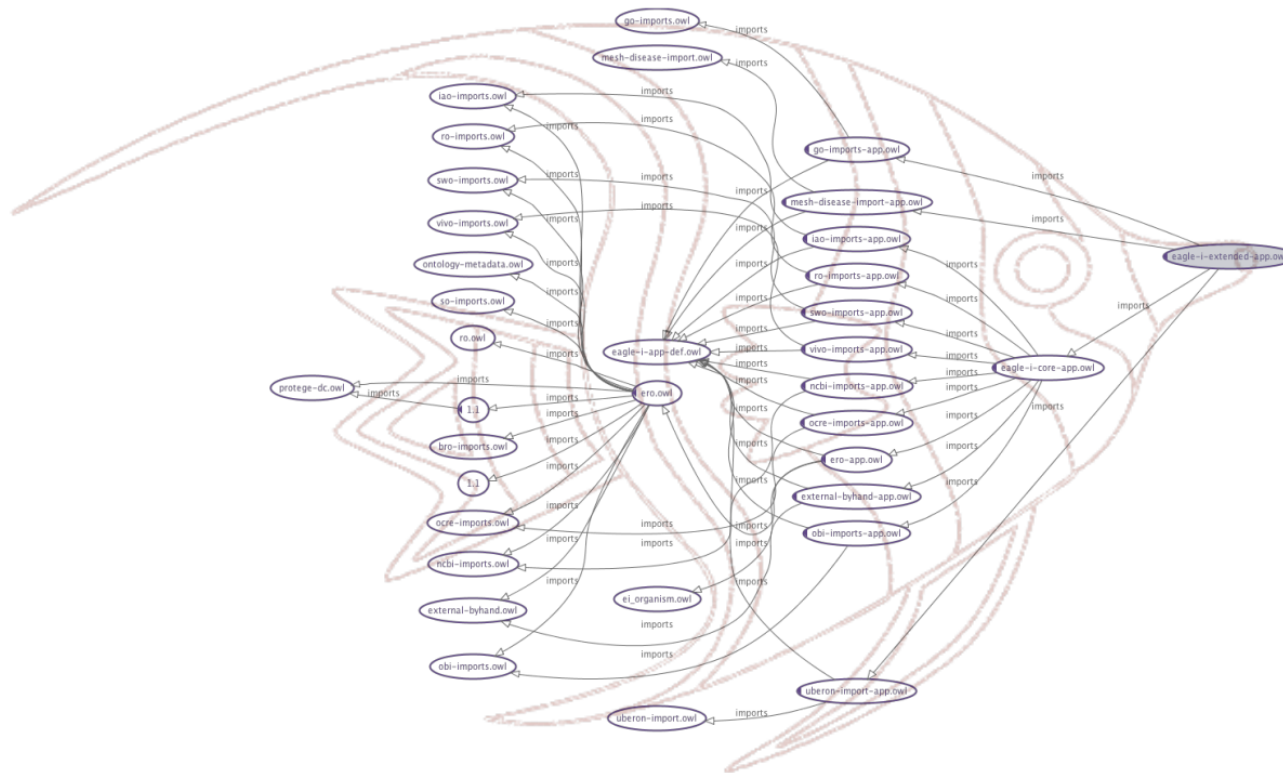


# Imports, MIREOT



Phenotype RCN



THE NATIONAL CENTER FOR  
BIOMEDICAL ONTOLOGY

Anatomy Ontologies for Evolutionary Biology and Genetics • Durham, NC • 2012

Contributors: Carlo Torniai, Melanie Courtot, Chris Mungall, Allen Xiang

# Importing ontologies

There is the need to reuse other existing OWL ontologies (as you've already seen) these ontologies can be:

- On the web
- In local files

# Importing ontologies

- Need to keep updated versions of imported ontologies
- Need ensure consistency for your developed ontologies
- In general, it's always better to work with local files (and just in case, have a local SVN repository to sync with current versions of other ontologies)

# owl:imports

- owl:imports statement references another OWL ontology containing definitions, whose meaning is considered to be part of the meaning of the importing ontology
- Each reference consists of a IRI specifying the location of the ontology that is to be imported
- Syntactically, owl:imports is a property with the class owl:Ontology as its domain and range
- The import statement is located in the ontology header
- The owl:imports statements are transitive, that is, if ontology A imports B, and B imports C, then A imports both B and C.

# owl:imports

## Ontology header

---

```
<owl:Ontology rdf:about="http://mysampleontology.org">  
  <owl:versionInfo>v 1.17 2003/02/26 12:56:51 mdean</owl:versionInfo>  
  <rdfs:comment>An example ontology</rdfs:comment>  
  <owl:imports rdf:resource="http://www.example.org/foo"/>  
</owl:Ontology>
```

# Challenges for importing other ontologies

- ***Computational overhead*** – some ontologies, such as NCBI Taxonomy or Foundational Model of Anatomy (FMA) are very large
- ***Alignment*** - Ontologies constructed using a different design or those not using BFO as upper-level ontology cannot be fully integrated
- ***Fluid development*** - Resources are *always* under development, need to think about which version to use, how to record that

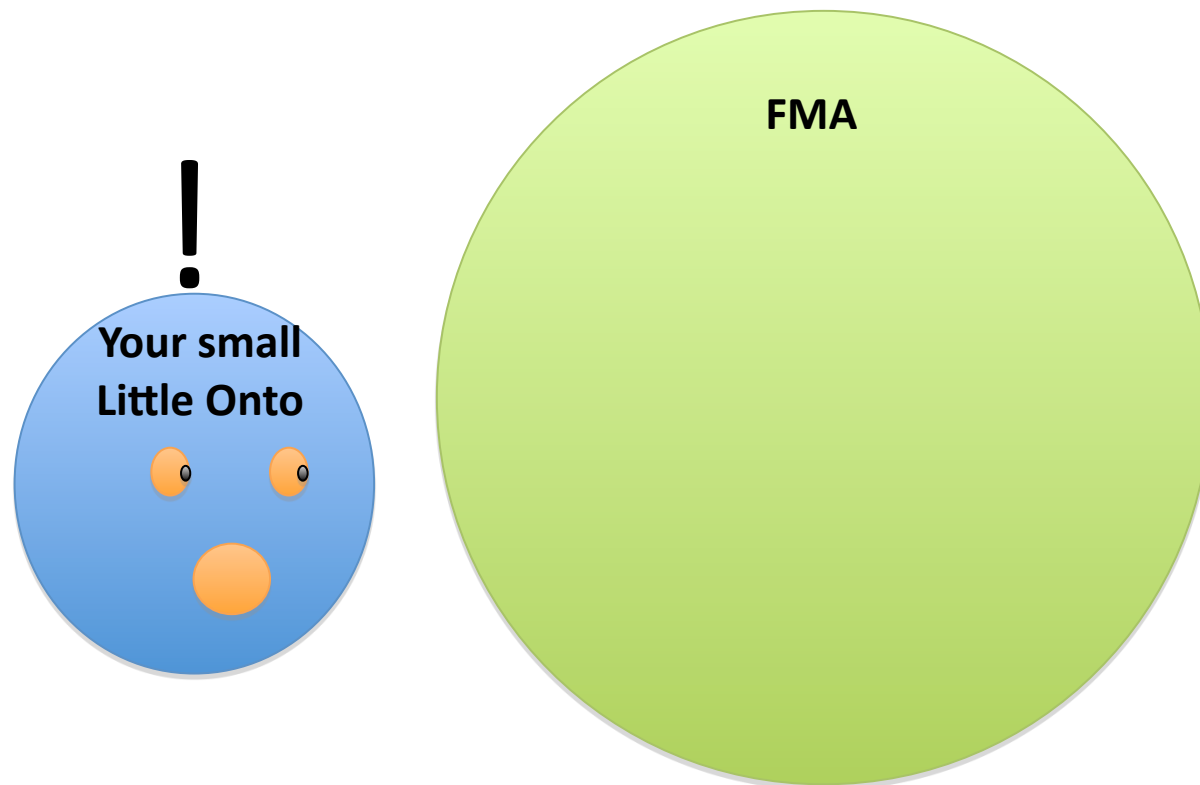
# Possible solutions

- Import the whole ontology
- Generate your own terms and reference other terms via xrefs
- Generate and import a module of an ontology: a complete subset of entities and axioms

# Full import

We can import whole resources

- only if full axiomatic interoperability (the holy grail of the ontology community)
- Large ontologies have huge overhead: current limitations in editing tools and reasoners

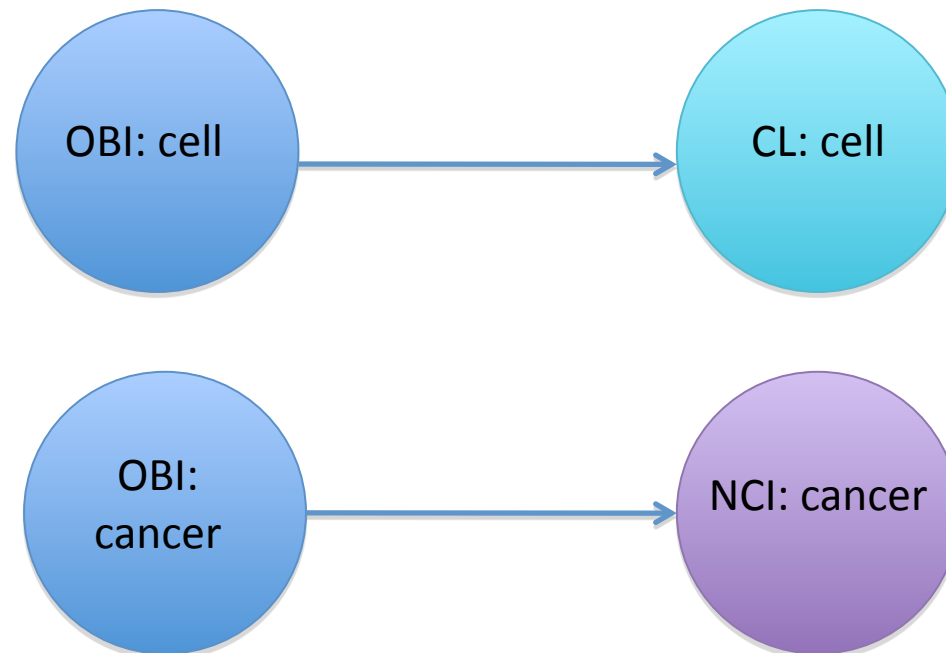




# Generate your own terms and use Xrefs

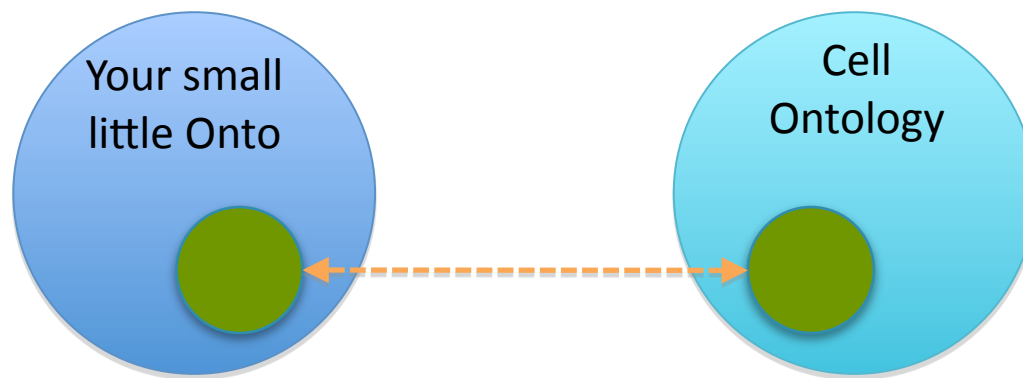
One can create our own terms and reference others

- Adding an annotation referencing the external ontology
- **But** this duplicates efforts, creates redundancy, doesn't comply with orthogonality principle from OBO Foundry, making data integration more difficult



# Generate an import module

- A module is a subset of the external ontology, containing classes and axioms, allowing “original” reasoning
- **But** it isn't trivial to create/ get the modules



# Dealing with imports in Protégé

- Tutorial document:  
**/thur/tutorials/imports\_tutorial.doc**
  - Install OWLviz plugin (if not installed already)
  - Review imports graph
  - Understanding imports in Protégé
    - Ontology libraries and Catalog file

## **Idea: Import only classes that are needed**

- Pro: We get around the problems of the other methods
- Con: one may lose complete inference

# MIREOT

- Minimal Information to Reference External  
Ontology Terms
  - Formal approach
  - Implementations

# Define the minimal information we need

- IRI of the class
- IRI of the source ontology
- Superclass in the recipient ontology

=> this ***minimal set*** allows us to unambiguously identify a term

# Additional information

- We may want to capture:
  - Label
  - Definition
  - Other annotations: adding “human-readable” information
  - SuperClasses: for example, NCBI taxonomy
  - ...

# Implementation

- Strategy:
  - Figure out how to automate as much as possible..  
Because if you edit owl....
- How to make it as easy as possible to enter, and maintain
- One (widely used) MIREOT implementation: OntoFox



# OntoFox

- OntoFox is a web-based system to support ontology reuse by applying the MIREOT principle
- Access it at: <http://ontofox.hegroup.org/>

# OntoFox tour

## 1. Data input using web forms:

Examples: [Example 1](#), [example 2](#), [example 3](#), [example 4](#), [example 5](#)

### (1) Select one ontology:

Please select an ontology

Or enter your favorite source ontology and SPARQL endpoint: [Example](#)

### (2) Term specification:

#### (a) Include low level source term URIs:

(One URI per line. To include all child terms of a source term (extract the whole branch), enter "includeAllChildren" in the line next to the source term)

Search a term:  Term ID:

#### (b) Include top level source term URIs and target direct superclass URIs (One URI per line, optional):

Search a term:  Term ID:

#### (c) Select a setting for retrieving intermediate source terms:

includeNoIntermediates

### (3) Annotation/Axiom Specification: Include source annotation URIs (One URI per line, optional):

Examples: [rdfs:label](#), [iao:preferredTerm](#), [iao:definition](#), [iao:alternative term](#), [oboInOwl:hasDefinition](#), [oboInOwl:hasSynonym](#), [owl:equivalentClass](#).

The default is no annotation to be assigned. Use [includeAllAnnotationProperties](#) to include all annotations. Use [includeAllAxioms](#) to include all annotations and other related axioms. Use [includeAllAxiomsRecursively](#) to include all axioms for the specified terms and the related terms recursively.

# Example

- Let's look for the Caro term 'cell' in Ontobee

## Class: cell

- Term IRI: [http://purl.obolibrary.org/obo/CARO\\_0000013](http://purl.obolibrary.org/obo/CARO_0000013)
- definition: **Anatomical structure that has as its parts a maximally connected cell compartment surrounded by a plasma membrane.** [database\_cross CARO:MAH]

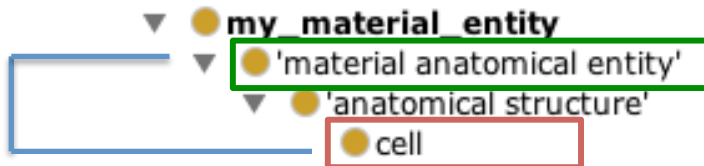
## Annotations

- id: **CARO:0000013**
- database\_cross\_reference: **GO:0005623**

## Class Hierarchy

```
Thing
+ anatomical entity
  + material anatomical entity
    + anatomical structure
      + multi-tissue structure
      + multi-cellular organism
      + acellular anatomical structure
      - organism subdivision
      + portion of tissue
      + compound organ
      + anatomical group
      - cell component
      - extraembryonic structure
      + multi-cell-component structure
      + cell
        - single cell organism
        - epithelial cell
```

# Example



(1) Select one ontology:

Common Anatomy Reference Ontology (CARO)

Or enter your favorite source ontology and SPARQL endpoint: [Example](#)

(2) Term specification:

(a) Include low level source term URIs:

(One URI per line. To include all child terms of a source term (extract the whole branch), enter

Search a term: cell (CARO)

Term ID: CARO\_0000013

[http://purl.obolibrary.org/obo/CARO\\_0000013](http://purl.obolibrary.org/obo/CARO_0000013) #cell (CARO)

(b) Include top level source term URIs and target direct superclass URIs (One URI per line, o

Search a term: material anatomical entity (CARO)

Term ID: CARO\_0000006

[http://purl.obolibrary.org/obo/CARO\\_0000006](http://purl.obolibrary.org/obo/CARO_0000006) #material anatomical entity  
subClassOf [http://purl.obolibrary.org/obo/my\\_material\\_entity](http://purl.obolibrary.org/obo/my_material_entity)

(c) Select a setting for retrieving intermediate source terms:

includeAllIntermediates

[http://purl.obolibrary.org/obo/my\\_material\\_entity](http://purl.obolibrary.org/obo/my_material_entity)

IS A CLASS IN YOUR TARGET ONTOLOGY i.e.  
THE ONTOLOGY THAT IS GOING TO IMPORT THIS FILE

# Example

- Let's look for the PATO term 'volume' in Ontobee
- What do you think I would get as result?

## Class: volume

- Term IRI: [http://purl.obolibrary.org/obo/PATO\\_0000918](http://purl.obolibrary.org/obo/PATO_0000918)
- definition: A 3-D extent quality inhering in a bearer by virtue of the bearer's amount of 3-dimensional space it occupies. [database\_cross\_reference: PATOC:GVG]

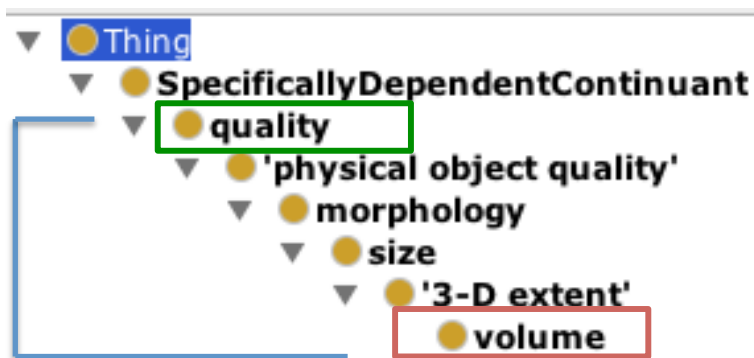
## Annotations

- id: PATO:0000918
- has\_obo\_namespace: quality
- in\_subset: scalar\_slim, attribute\_slim

## Class Hierarchy

```
Thing
+ quality
  + physical object quality
    + morphology
      + size
        + 3-D extent
          + volume
            + increased volume
            - molar volume
            + decreased volume
            - specific volume
```

# Example



## (2) Term specification:

(a) Include low level source term URIs:

(One URI per line. To include all child terms of a source term (extract the whole branch), enter "incl

Search a term:  Term ID:

[http://purl.obolibrary.org/obo/PATO\\_0000918#volume](http://purl.obolibrary.org/obo/PATO_0000918#volume)

(b) Include top level source term URIs and target direct superclass URIs (One URI per line, optional

Search a term:  Term ID:

[http://purl.obolibrary.org/obo/PATO\\_0000001#quality](http://purl.obolibrary.org/obo/PATO_0000001#quality)

subClassOf <http://www.ifomis.org/bfo/1.1/snap#SpecificallyDependentContinuant>

<http://www.ifomis.org/bfo/1.1/snap#SpecificallyDependentContinuant>  
IS A CLASS IN YOUR TARGET ONTOLOGY i.e.  
THE ONTOLOGY THAT IS GOING TO IMPORT THIS FILE

# Example

- What should I get with this settings?

(1) Select one ontology:

Phenotypic quality (PATO)

Or enter your favorite source ontology and SPARQL endpoint: [Example](#)

(2) Term specification:

(a) Include low level source term URIs:

(One URI per line. To include all child terms of a source term (extract the whole branch), enter "includeAllChildren" in the line next to the source term)

Search a term:  Term ID:

`http://purl.obolibrary.org/obo/PATO_0000918 #volume`

(b) Include top level source term URIs and target direct superclass URIs (One URI per line, optional):

Search a term:  Term ID:

`http://purl.obolibrary.org/obo/PATO_0000001 #quality`  
`subClassOf http://www.ifomis.org/bfo/1.1/snap#SpecificallyDependentContinuant`

(c) Select a setting for retrieving intermediate source terms:

includeNoIntermediates

# Example



(1) Select one ontology:

Phenotypic quality (PATO) ▾

Or enter your favorite source ontology and SPARQL endpoint: [Example](#)

(2) Term specification:

(a) Include low level source term URIs:

(One URI per line. To include all child terms of a source term (extract the whole branch), enter "includeAllChildren")

Search a term:  ▾ Term ID:

`http://purl.obolibrary.org/obo/PATO_0000918 #volume`

(b) Include top level source term URIs and target direct superclass URIs (One URI per line, optional):

Search a term:  ▾ Term ID:

`http://purl.obolibrary.org/obo/PATO_0000001 #quality`  
`subClassOf http://www.ifomis.org/bfo/1.1/snap#SpecificallyDependentContinuant`

(c) Select a setting for retrieving intermediate source terms:

includeNoIntermediates ▾



# OntoFox Tutorial

- Go on the tutorial file:

**`material_for_course/thur/tutorials/ontofox_tutorial.docx`**

# References

- MIREOT:
- <http://obi-ontology.org/page/MIREOT>

# Backup

# Select Ontology

(1) Select one ontology:

Please select an ontology

Or enter your favorite source ontology and SPARQL endpoint: [Example](#)

(2) Term specification:

(a) Include low level source term URIs:

(One URI per line. To include all child terms of a source term (extract the whole branch), enter "includeAllChild"

Search a term:  Term ID:

(b) Include top level source term URIs and target direct superclass URIs (One URI per line, optional):

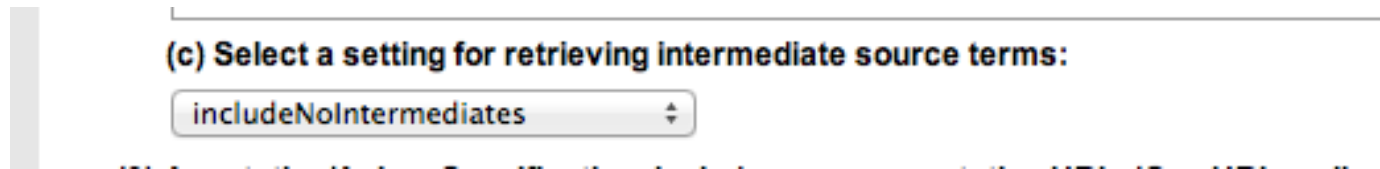
Search a term:  Term ID:

(c) Select a setting for retrieving intermediate source terms:

includeNoIntermediates

- The source ontology you want to retrieve the term(s) from

# Term specification: intermediate terms



**(c) Setting for retrieving intermediate source terms:** Three options are available for retrieving intermediate terms:

- (a) **includeNoIntermediates**: no intermediate source terms are retrieved
- (b) **includeComputedIntermediates**: Computed intermediate source terms include those intermediate terms that are the closest ancestors of more than one low level source terms. Those intermediate terms that have only one parent term and one child term each are removed. This setting provides an option to get less intermediate ontology terms than that with the setting 'includeAllIntermediates' and still fulfills many users' requirement.
- (c) **includeAllIntermediates**: All intermediate source terms are retrieved.

# Term specification

## (2) Term specification:

(a) Include low level source term URIs:

(One URI per line. To include all child terms of a source term (extract the whole branch), enter "includeAllChildren" in the line next to the source term)

Search a term:  ▼ Term ID:

(b) Include top level source term URIs and target direct superclass URIs (One URI per line, optional):

Search a term:  ▼ Term ID:

- **(a) Low level source term URIs:** The URIs of low level terms from source ontologies.
- **(b) Top level source term URIs and target direct superclass URIs:** The URIs of top level terms from source ontologies and their direct superclass URIs from a target ontology (i.e., the ontology that will import the terms from the source ontologies). The top level source term URI can be the same as the low level source term URI

# Annotation/Axioms Specification

(3) Annotation/Axiom Specification: Include source annotation URIs (One URI per line, optional):

Examples: [rdfs:label](#), [iao:preferredTerm](#), [iao:definition](#), [iao:alternative term](#), [obolOwl:hasDefinition](#), [obolOwl:hasSynonym](#), [owl:equivalentClass](#).

The default is no annotation to be assigned. Use [includeAllAnnotationProperties](#) to include all annotations. Use [includeAllAxioms](#) to include all annotations and other related axioms. Use [includeAllAxiomsRecursively](#) to include all axioms for the specified terms and the related terms recursively.



- **includeAllAnnotationProperties**: By default, if no annotation URI is assigned, no annotations associated with a specific ontology term will be fetched. To include all possible annotations, you can put "**includeAllAnnotationProperties**" on one line, and all the annotations associated with a specific ontology term will be fetched.
- **includeAllAxioms**: To include all possible annotations and related axioms for a specified term(s), you can put "**includeAllAxioms**" on one line, and all the axioms associated with a specific ontology term(s) will be fetched.
- **includeAllAxiomsRecursively**: To include all possible annotations and related axioms for a specified term(s) and its associated terms recursively, you can enter "**includeAllAxiomsRecursively**" on one line. Note: "**includeNoIntermediates**" and "**includeComputedIntermediates**" have higher priority and will override "**includeAllAxiomsRecursively**".