skosprovider_getty Documentation

Release 0.5.1

Flanders Heritage Agency

Oct 06, 2020

Contents

CHAPTER 1

Introduction

This library offers an implementation of the `skosprovider.providers.VocabularyProvider` interface based on the Getty Vocabularies. It reduces the complex vocabularies like *AAT* and *TGN* to a basic *SKOS* version of them.

Supported Getty thesauri:

- The *Art & Architecture Thesaurus (AAT)* by use of the *skosprovider_getty.providers.AATProvider*.
- The *Getty Thesaurus of Geographic Names (TGN)* by use of the *skosprovider_getty.providers.TGNProvider*.
- The *Union List of Artist Names (ULAN)* by use of the *skosprovider_getty.providers.ULANProvider*.

CHAPTER 2

Installation

To be able to use this library you need to have a modern version of Python installed. Currently we're supporting versions 2.7, 3.3 and 3.4 of Python.

This easiest way to install this library is through pip or easy install:

```
$ pip install skosprovider_getty
```

This will download and install `skosprovider_getty` and a few libraries it depends on.

CHAPTER 3

Using the providers

3.1 Using AATProvider

The *AATProvider* is a provider for the *AAT*. It's use is identical to all other SKOSProviders.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
'''
This script demonstrates using the AATProvider to get the concept of
Churches.
'''

from skosprovider_getty.providers import AATProvider

aat = AATProvider(metadata={'id': 'AAT'})

churches = aat.get_by_id(300007466)

lang = ['en', 'nl', 'es', 'de', 'fr']

print('Label per language')
print('------------------')
for l in lang:
    label = churches.label(l)
    print(l + ' --> ' + label.language + ': ' + label.label + ' [' + label.type + ']')

print('All Labels')
print('----------')
for l in churches.labels:
   print(l.language + ': ' + l.label + ' [' + l.type + ']')

print('All Notes')
print('---------')
for n in churches.notes:
```

(continues on next page)

(continued from previous page)

```
    print(n.language + ': ' + n.note + ' [' + n.type + ']')
```

3.2 Using TGNProvider

The *TGNProvider* is a provider for the *TGN*. It's use is identical to all other SKOSProviders.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
'''
This script demonstrates using the TGNProvider to get the concept of
Flanders.
'''

from skosprovider_getty.providers import TGNProvider

aat = TGNProvider(metadata={'id': 'TGN'})

flanders = aat.get_by_id(7018236)

lang = ['en', 'nl', 'es', 'de', 'fr']

print('Label per language')
print('------------------')
for l in lang:
    label = flanders.label(l)
    print(l + ' --> ' + label.language + ': ' + label.label + ' [' + label.type + ']')

print('Labels')
print('------')
for l in flanders.labels:
   print(l.language + ': ' + l.label + ' [' + l.type + ']')

print('Notes')
print('-----')
for n in flanders.notes:
    print(n.language + ': ' + n.note + ' [' + n.type + ']')
```

3.3 Finding concepts or collections

See the *skosprovider_getty.providers.GettyProvider.find()* method for a detailed description of how this works.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
'''
This script demonstrates using the AATProvider to find the concepts that are a member␣
↪of a collection with 'church' in their label
'''

from skosprovider_getty.providers import AATProvider
```

(continues on next page)

(continued from previous page)

```
results = AATProvider({'id': 'AAT', 'default_language': 'nl'}).find({'label': 'church
↪', 'type': 'concept', 'collection': {'id': '300007466', 'depth': 'all'}})

print('Results')
print('------')
for result in results:
    print(result)
```

3.4 Using expand()

The expand methods return the id's of all the concepts that are narrower concepts of a certain concept or collection.

See the *skosprovider_getty.providers.GettyProvider.expand()* method for a detailed description of how this works.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
'''
This script demonstrates using the AATProvider to expand a collection
'''

from skosprovider_getty.providers import AATProvider

results = AATProvider({'id': 'AAT', 'default_language': 'nl'}).expand('300007466')

print('Results')
print('------')
for result in results:
    print(result)
```

CHAPTER 4

Development

Skosprovider_getty is being developed by the Flanders Heritage Agency.

Since we place a lot of importance on code quality, we expect to have a good amount of code coverage present and run frequent unit tests. All commits and pull requests will be tested with Travis-ci. Code coverage is being monitored with Coveralls.

Locally you can run unit tests by using pytest or tox. Running pytest manually is good for running a distinct set of unit tests. For a full test run, tox is preferred since this can run the unit tests against multiple versions of python.

```
# Setup for development
$ python setup.py develop
# Run unit tests for all environments
$ tox
# No coverage
$ py.test
# Coverage
$ py.test --cov skosprovider_getty --cov-report term-missing tests
# Only run a subset of the tests
$ py.test skosprovider_getty/tests/test_providers.py
```

Please provide new unit tests to maintain 100% coverage. If you send us a pull request and this build doesn't function, please correct the issue at hand or let us know why it's not working.

CHAPTER 5

API Documentation

5.1 Providers module

This module contains classes that implement `skosprovider.providers.VocabularyProvider` against the LOD version of the Getty Vocabularies (AAT, TGN and ULAN).

Note:

At initialisation, the Getty providers will search which gvp-classes of the gvp-ontology are a subclass of skos-classes. This can cause a time delay of several seconds at startup.

class skosprovider_getty.providers.**AATProvider**(*metadata*, ***kwargs*)
The Art & Architecture Thesaurus Provider A provider that can work with the GETTY AAT rdf files of http://vocab.getty.edu/aat

class skosprovider_getty.providers.**GettyProvider**(*metadata*, ***kwargs*)
A provider that can work with the GETTY rdf files of http://vocab.getty.edu/

expand(*id*)

Expand a concept or collection to all it's narrower concepts. If the id passed belongs to a `skosprovider.skos.Concept`, the id of the concept itself should be include in the return value.

Parameters **id** (*str*) – A concept or collection id.

Returns A `lst` of id's. Returns false if the input id does not exists

find(*query*, ***kwargs*)
Find concepts that match a certain query.

Currently query is expected to be a dict, so that complex queries can be passed. You can use this dict to search for concepts or collections with a certain label, with a certain type and for concepts that belong to a certain collection.

```
# Find anything that has a label of church.
provider.find({'label': 'church'}

# Find all concepts that are a part of collection 5.
provider.find({'type': 'concept', 'collection': {'id': 5})

# Find all concepts, collections or children of these
# that belong to collection 5.
provider.find({'collection': {'id': 5, 'depth': 'all'})
```

Parameters **query** – A dict that can be used to express a query. The following keys are permitted:

- *label*: Search for something with this label value. An empty label is equal to searching for all concepts.
- *type*: Limit the search to certain SKOS elements. If not present *all* is assumed:
  - *concept*: Only return `skosprovider.skos.Concept` instances.
  - *collection*: Only return `skosprovider.skos.Collection` instances.
  - *all*: Return both `skosprovider.skos.Concept` and `skosprovider.skos.Collection` instances.
- *collection*: Search only for concepts belonging to a certain collection. This argument should be a dict with two keys:
  - *id*: The id of a collection. Required.
  - *depth*: Can be *members* or *all*. Optional. If not present, *members* is assumed, meaning only concepts or collections that are a direct member of the collection should be considered. When set to *all*, this method should return concepts and collections that are a member of the collection or are a narrower concept of a member of the collection.

Returns

A `lst` of concepts and collections. Each of these is a dict with the following keys:

- id: id within the conceptscheme
- uri: *uri* of the concept or collection
- type: concept or collection
- label: A label to represent the concept or collection. It is determined by looking at the ***kwargs* parameter, the default language of the provider and finally falls back to *en*.

get_all(***kwargs*)
Not supported: This provider does not support this. The amount of results is too large

get_by_id(*id*, *change_notes=False*)
Get a `skosprovider.skos.Concept` or `skosprovider.skos.Collection` by id

Parameters **id** (*(str)*) – integer id of the `skosprovider.skos.Concept` or `skosprovider.skos.Concept`

Returns corresponding `skosprovider.skos.Concept` or `skosprovider.skos.Concept`. Returns None if non-existing id

get_by_uri(*uri*, *change_notes=False*)
Get a `skosprovider.skos.Concept` or `skosprovider.skos.Collection` by uri

Parameters **uri** (*(str)*) – string uri of the skosprovider.skos.Concept or skosprovider.skos.Concept

Returns corresponding skosprovider.skos.Concept or skosprovider.skos.Concept. Returns None if non-existing id

get_children_display(*id*, ***kwargs*)
Return a list of concepts or collections that should be displayed under this concept or collection.

Parameters **id** (*str*) – A concept or collection id.

Returns A lst of concepts and collections.

get_top_concepts(***kwargs*)
Returns all concepts that form the top-level of a display hierarchy.

Returns A lst of concepts.

get_top_display(***kwargs*)
Returns all concepts or collections that form the top-level of a display hierarchy.

Returns A lst of concepts and collections.

class skosprovider_getty.providers.**TGNProvider**(*metadata*, ***kwargs*)
The Getty Thesaurus of Geographic Names A provider that can work with the GETTY TGN rdf files of http://vocab.getty.edu/tgn

class skosprovider_getty.providers.**ULANProvider**(*metadata*, ***kwargs*)
Union List of Artist Names

A provider that can work with the GETTY ULAN rdf files of http://vocab.getty.edu/ulan

5.2 Utility module

This module contains utility functions for skosprovider_getty.

class skosprovider_getty.utils.**SubClassCollector**(*namespace*)
A utility class to collect all the subclasses of a certain Class from an ontology file.

collect_subclasses(*clazz*)
Collect all subclasses for a class and override the registered classes.

Since this requires fetching ontology files, it might take a while.

Parameters **clazz** – An RDF class

Returns A list of all subclasses, including the original class.

get_subclasses(*clazz*)
Get all registered subclasses for a class.

Parameters **clazz** – An RDF class

Returns A list of all subclasses, including the original class.

skosprovider_getty.utils.**conceptscheme_from_uri**(*conceptscheme_uri*, ***kwargs*)
Read a SKOS Conceptscheme from a *URI*

Parameters **conceptscheme_uri** (*string*) – URI of the conceptscheme.

Return type skosprovider.skos.ConceptScheme

skosprovider_getty.utils.**uri_to_graph**(*uri*, ***kwargs*)

Parameters **uri** (*string*) – *URI* where the RDF data can be found.

Return type rdflib.Graph or *False* if the URI does not exist

Raises **skosprovider.exceptions.ProviderUnavailableException** – if the getty.edu services are down

CHAPTER 6

History

6.1 0.5.1 (2020-10-06)

- Prevent get_by_uri erroring on non-Getty URI's (#77)
- Remove reference to nose.collector
- Remove pyup integration

6.2 0.5.0 (2020-08-06)

- Compatibile with SkosProvider 0.7.0. (#59)
- Prevent unnecessary loading of conceptschemes. (#56)
- Update to RDFlib 5.0.0 (#69)
- Supports Python 2.7, 3.6, 3.7 and 3.8. Last version to support Python 2.

6.3 0.4.2 (2017-09-06)

- Really stop loading the conceptscheme while initialising the provider.

6.4 0.4.1 (2017-09-06)

- Stop loading the conceptscheme while initialising the provider.

6.5 0.4.0 (2017-07-15)

- Stop collecting SKOS Concept and Collection subclasses. They are now included in the code base since they seem to have become rather stable and this reduces the startup time of the provider significantly. (#28)
- Add support for python 3.6 when testing.

6.6 0.3.1 (2016-09-14)

- Handle a bug with private language tags. Currently not recognised by the language_tags library. The Getty services do use them. When encountered, we fall back to the undeterminded language. (#26, #27)

6.7 0.3.0 (2016-08-11)

- Upgrade to skosprovider 0.6.0. (#13)
- Add support for the ULAN vocabulary. (#22)
- Add support for sorting. (#24)
- Allow configuring the requests session in use. (#25)

6.8 0.2.1 (2015-03-10)

- Introduce language support. Until now it was impossible to pass in a language parameter to certain methods. This was not only a missing feature, but also a bug since the VocabularyProvider interface requires that a client can pass in extra keywords. (#16)
- iso-thes:superordinates get fetched from the SPARQL store. (#17)
- All network requests now go through requests. (#13)
- Some documentation improvements. (#15)

6.9 0.2.0 (2014-12-22)

- Compatibile with SkosProvider 0.5.x.
- Now uses the IANA language code *und* for labels in an unknown language.
- Now throws a ProviderUnavailableException when the Getty vocab services can't be reached.
- Handle superordinates and subordinate arrays.
- Error handling and bugfixes.

6.10 0.1.0 (2014-09-30)

- Initial version
- Contains providers for AAT and TGN vocabularies.

- Compatible with SkosProvider 0.3.0.

CHAPTER 7

Glossary

AAT The Art & Architecture Thesaurus. A vocabulary provided by the *Getty*.

Getty The J. Paul Getty Trust is a cultural and philanthropic institution dedicated to critical thinking in the presentation, conservation, and interpretation of the world's artistic legacy. Through the collective and individual work of its constituent Programs—Getty Conservation Institute, Getty Foundation, J. Paul Getty Museum, and Getty Research Institute—it pursues its mission in Los Angeles and throughout the world, serving both the general interested public and a wide range of professional communities with the conviction that a greater and more profound sensitivity to and knowledge of the visual arts and their many histories are crucial to the promotion of a vital and civil society. More information: http://www.getty.edu

RDF Resource Description Framework. A very flexible model for data definition organised around *triples*. These triples forms a directed, labeled graph, where the edges represent the named link between two resources, represented by the graph nodes.

SKOS Simple Knowledge Organization System. An general specification for Knowledge Organisation Systems (thesauri, word lists, authority files, . . .) that is commonly serialised as *RDF*.

TGN The Getty Thesaurus of Geographic Names. A vocabulary provided by the *Getty*.

URI A *Uniform Resource Identifier*.

URN A URN is a specific form of a *URI*.

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

S

Index