# Pyramid Skosprovider Documentation

***Release 0.5.0***

**Koen Van Daele**

March 04, 2015

Contents

This library helps to integrate skosprovider in a Pyramid application.

CHAPTER 1

# Installation

To install *pyramid_skosprovider*, use pip

```
pip install pyramid_skosprovider
```

To activate *pyramid_skosprovider*, just include it.

```
config = Configurator()
config.include('pyramid_skosprovider')
```

This will create a `skosprovider.registry.Registry` and add it to the pyramid application registry.

CHAPTER 2

# Usage

To get a `skosprovider.registry.Registry` instance, call `pyramid_skosprovider.get_skos_registry()` with the current application registry.

Eg. in a view:

```
from pyramid_skosprovider import get_skos_registry

def my_view(request):
    skos = get_skos_registry(request.registry)
    providers = skos.get_providers()
    # ...
```

Alternatively you can get the registry as an attribute of a pyramid request:

```
from pyramid_skosprovider import get_skos_registry

def my_view(request):
    skos = request.skos_registry
    providers = skos.get_providers()
    # ...
```

For a real-world example of an integration of *pyramid_skosprovider* in a *Pyramid* application, have a look at Atramhasis, a SKOS vocabulary editor partially built upon this library.

CHAPTER 3

# Service Documentation

This library takes your skosproviders and makes them available as REST services. The pyramid_skosprovider serves JSON as a REST service so it can be used easily inside a AJAX webbrowser call or by an external program.

The following API can be used by clients:

**GET /uris/{uri}**
Find more information on a certain *URI*. This can map to eiter a concept, collection or conceptscheme that is known by the current SKOS registry.

**Example request**:

```
GET /uris/urn:x-skosprovider:trees HTTP/1.1
Host: localhost:6543
Accept: application/json
```

**Example response**:

```
HTTP/1.1 200 OK
Content-Type:  application/json; charset=UTF-8

{
    "id": "TREES",
    "uri": "urn:x-skosprovider:trees",
    "type": "concept_scheme"
}
```

**Example request**:

```
GET /uris/http://python.com/trees/larch HTTP/1.1
Host: localhost:6543
Accept: application/json
```

**Example response**:

```
HTTP/1.1 200 OK
Content-Type:  application/json; charset=UTF-8

{
    "id": "1",
    "uri": "http://python.com/trees/larch",
    "type": "concept",
    "concept_scheme": {
        "id": "TREES",
        "uri": "urn:x-skosprovider:trees"
```

```
        }
    }
```

**Status Codes**

- 200 OK – The URI maps to something known by pyramid_skosprovider, either a conceptscheme, a concept or collection.
- 404 Not Found – The URI can't be found by pyramid_skosprovider.

**GET /c**

Search for concepts or collections, no matter what scheme they're a part of.

Although it is possible to search a single conceptscheme with just this endpoint, for performance reasons it is advised to use `GET /conceptschemes/{scheme_id}/c`.

**Example request**:

```
GET /c HTTP/1.1
Host: localhost:6543
Accept: application/json
```

**Example response**:

```
HTTP/1.1 200 OK
Content-Range:  items 0-2/232
Content-Type:  application/json; charset=UTF-8

[
    {
        "id": "1",
        "uri": "urn:x-skosprovider:TREES:1",
        "type": "concept",
        "label": "De Lariks"
    }, {
        "id": "2",
        "uri": "urn:x-skosprovider:TREES:2",
        "type": "concept",
        "label": "De Paardekastanje"
    }, {
        "id": 3,
        "uri": "urn:x-skosprovider:TREES:3",
        "type": "collection",
        "label": "Bomen per soort"
    }
]
```

**Example request**:

```
GET /c?type=concept&providers.subject=external&sort=uri HTTP/1.1
Host: localhost:6543
Accept: application/json
```

**Query Parameters**

- **type** – Define if you want to show concepts or collections. Leave blank to show both.
- **mode** – Allows for special processing mode for dijitFilteringSelect. Makes it possible to use wildcards in the label parameter.

- **label** – Shows all concepts and collections that have this search string in one of their labels.
- **sort** – Define if you want to sort the results by a given field. Otherwise items are returned in an indeterminate order. Prefix with '+' to sort ascending, '-' to sort descending. eg. `?sort=-label` to sort all results descending by label.
- **providers.ids** – A comma separated list of concept scheme id's. The query will only be passed to the providers with these id's. eg. `?providers.ids=TREES, PARROTS` will only list concepts from these two providers.
- **providers.subject** – A subject can be registered with a skosprovider in the registry. Adding this search parameter means that the query will only be passed on to providers that have been tagged with this subject. Eg. `?providers.subject=external` to only query the providers that have been marked with the subject *external*.

**Request Headers**

- Range – Can be used to request a certain set of results. eg. `items=0-24` requests the first 25 results.

**Response Headers**

- Content-Range – Tells the client what set of results is being returned eg. `items=0-24/306` means the first 25 out of 306 results are being returned.

**Status Codes**

- 200 OK – The concepts in this conceptscheme were found.

**GET /conceptschemes**

Get all registered conceptschemes.

**Example request**:

```
GET /conceptschemes HTTP/1.1
Host: localhost:6543
Accept: application/json
```

**Example response**:

```
HTTP/1.1 200 OK
Content-Type:  application/json; charset=UTF-8
Date:  Mon, 14 Apr 2014 14:42:34 GMT

[
    {
        "id": "TREES",
        "uri": "urn:x-skosprovider:trees",
        "label": "Different types of trees."
    }
]
```

**Status Codes**

- 200 OK – The list of conceptschemes was found.

**GET /conceptschemes/{scheme_id}**

Get information about a concept scheme.

**Example request**:

```
GET /conceptschemes/TREES
Host: localhost:6543
Accept: application/json
```

**Example response**:

```
HTTP/1.1 200 OK
Content-Length:  15
Content-Type:  application/json; charset=UTF-8
Date:  Mon, 14 Apr 2014 14:45:37 GMT
Server:  waitress

{
    "id": "TREES",
    "uri": "urn:x-skosprovider:trees",
    "label": "Different types of trees.",
    "labels": [
        {"type": "prefLabel", "language": "en", "label": "Different types of trees."},
        {"type": "prefLabel", "language": "nl", "label": "Verschillende soorten bomen."}
    ]
}
```

**Example request**:

-.. sourcecode:: http

GET /conceptschemes/PLANTS Host: localhost:6543 Accept: application/json

**Example response**:

```
HTTP/1.1 404 Not Found
Content-Length:  775
Content-Type:  text/html; charset=UTF-8
Date:  Tue, 15 Apr 2014 20:32:52 GMT
Server:  waitress
```

**Status Codes**

- 200 OK – The conceptscheme was found.
- 404 Not Found – The conceptscheme was not found.

**GET /conceptschemes/{scheme_id}/topconcepts**

Get all top concepts in a certain conceptscheme. These are all the concepts in the conceptscheme that have no broader concept.

**Example request**:

```
GET /conceptschemes/TREES/topconcepts
Host: localhost:6543
Accept: application/json
```

**Example response**:

```
HTTP/1.1 200 OK
Content-Type:  application/json; charset=UTF-8
Date:  Mon, 14 Apr 2014 14:47:33 GMT
Server:  waitress

[
    {
```

```
        "id": "1",
        "uri": "urn:x-skosprovider:TREES:1",
        "type": "concept",
        "label": "De Lariks"
    }, {
        "id": "2",
        "uri": "urn:x-skosprovider:TREES:2",
        "type": "concept",
        "label": "De Paardekastanje"
    }
]
```

**Status Codes**

- 200 OK – The topconcepts in this conceptscheme were found.
- 404 Not Found – The conceptscheme was not found.

**GET /conceptschemes/{scheme_id}/displaytop**

Get the top of a display hierarchy. Depending on the underlying provider this will be a list of Concepts and Collections.

**Example request**:

```
GET /conceptschemes/TREES/displaytop
Host: localhost:6543
Accept: application/json
```

**Example response**:

```
HTTP/1.1 200 OK
Content-Type:  application/json; charset=UTF-8
Date:  Mon, 14 Apr 2014 14:47:33 GMT
Server:  waitress

[
    {
        "id": "1",
        "uri": "urn:x-skosprovider:TREES:1",
        "type": "concept",
        "label": "De Lariks"
    }, {
        "id": "2",
        "uri": "urn:x-skosprovider:TREES:2",
        "type": "concept",
        "label": "De Paardekastanje"
    }
]
```

**Status Codes**

- 200 OK – The concepts and collections were found.
- 404 Not Found – The conceptscheme was not found.

**GET /conceptschemes/{scheme_id}/c**

Search for concepts or collections in a scheme.

**Example request**:

```
GET /conceptschemes/TREES/c
Host: localhost:6543
Accept: application/json
```

**Example response**:

```
HTTP/1.1 200 OK
Content-Length:  117
Content-Range:  items 0-2/3
Content-Type:  application/json; charset=UTF-8
Date:  Mon, 14 Apr 2014 14:47:33 GMT
Server:  waitress

[
    {
        "id": "1",
        "uri": "urn:x-skosprovider:TREES:1",
        "type": "concept",
        "label": "De Lariks"
    }, {
        "id": "2",
        "uri": "urn:x-skosprovider:TREES:2",
        "type": "concept",
        "label": "De Paardekastanje"
    }, {
        "id": 3,
        "uri": "urn:x-skosprovider:TREES:3",
        "type": "collection",
        "label": "Bomen per soort"
    }
]
```

**Example request**:

```
GET /conceptschemes/PLANTS/c
Host: localhost:6543
Accept: application/json
```

**Example response**:

```
HTTP/1.1 404 Not Found
Content-Length:  775
Content-Type:  text/html; charset=UTF-8
Date:  Tue, 15 Apr 2014 20:32:52 GMT
Server:  waitress
```

**Query Parameters**

- **type** – Define if you want to show concepts or collections. Leave blank to show both.
- **mode** – Allows for special processing mode for dijitFilteringSelect. Makes it possible to use wildcards in the label parameter.
- **label** – Shows all concepts and collections that have this search string in one of their labels.
- **collection** – Get information about the content of a collection. Expects to be passed an id of a collection in this scheme. Will restrict the search to concepts or collections that are a member of this collection or a narrower concept of a member.
- **sort** – Define if you want to sort the results by a given field. Otherwise items are returned

in an indeterminate order. Prefix with '+' to sort ascending, '-' to sort descending. eg. `?sort=-label` to sort all results descending by label.

**Request Headers**

- Range – Can be used to request a certain set of results. eg. `items=0-24` requests the first 25 results.

**Response Headers**

- Content-Range – Tells the client was set of results is being returned eg. `items=0-24/306` means the first 25 out of 306 results are being returned.

**Status Codes**

- 200 OK – The concepts in this conceptscheme were found.
- 404 Not Found – The conceptscheme was not found.

**GET /conceptschemes/{scheme_id}/c/{c_id}**

Get information about a concept or collection.

**Example request**:

```
GET /conceptschemes/TREES/c/1
Host: localhost:6543
Accept: application/json
```

**Example response**:

```
HTTP/1.1 200 OK
Content-Length:  316
Content-Type:  application/json; charset=UTF-8
Date:  Mon, 14 Apr 2014 14:49:27 GMT
Server:  waitress

{
    "broader": [],
    "narrower": [],
    "notes": [
        {"note": "A type of tree.", "type": "definition", "language": "en"}
    ],
    "labels": [
        {"type": "prefLabel", "language": "en", "label": "The Larch"},
        {"type": "prefLabel", "language": "nl", "label": "De Lariks"}
    ],
    "type": "concept",
    "id": "1",
    "uri": "urn:x-skosprovider:TREES:1",
    "related": [],
    "label": "The Larch",
    "matches": {
        "close": [
            'http://id.python.org/different/types/of/trees/nr/1/the/larch'
        ]
    },
    concept_scheme: {
        'uri': 'urn:x-foo:bar
    }
}
```

**Example request**:

```
GET /conceptschemes/TREES/c/4
Host: localhost:6543
Accept: application/json
```

**Example response**:

```
HTTP/1.1 404 Not Found
Content-Length:  775
Content-Type:  text/html; charset=UTF-8
Date:  Tue, 15 Apr 2014 20:06:12 GMT
Server:  waitress
```

**Status Codes**

- 200 OK – The concept was found in the conceptscheme.
- 404 Not Found – The concept was not found in the conceptscheme or the conceptscheme was not found.

**GET /conceptschemes/{scheme_id}/c/{c_id}/displaychildren**
Get a list of Collections and Concepts that should be displayed as children of this Concept or Collection.

**Example request**:

```
GET /conceptschemes/TREES/c/3/displaychildren
Host: localhost:6543
Accept: application/json
```

**Example response**:

```
HTTP/1.1 200 OK
Content-Type:  application/json; charset=UTF-8
Date:  Mon, 14 Apr 2014 14:49:27 GMT
Server:  waitress

[
    {
        "id": "1",
        "uri": "urn:x-skosprovider:TREES:1",
        "type": "concept",
        "label": "De Lariks"
    }, {
        "id": "2",
        "uri": "urn:x-skosprovider:TREES:2",
        "type": "concept",
        "label": "De Paardekastanje"
    }
]
```

**Status Codes**

- 200 OK – The concept was found in the conceptscheme.
- 404 Not Found – The concept was not found in the conceptscheme or the conceptscheme was not found.

CHAPTER 4

# API Documentation

## 4.1 Pyramid_skosprovider module

pyramid_skosprovider.**get_skos_registry**(*registry*)
Get the skosprovider.registry.Registry attached to this pyramid application.

**Return type** skosprovider.registry.Registry

## 4.2 Utils module

pyramid_skosprovider.utils.**collection_adapter**(*obj*, *request*)
Adapter for rendering a skosprovider.skos.Collection to json.

**Parameters** **obj** (*skosprovider.skos.Collection*) – The collection to be rendered.

**Return type** dict

pyramid_skosprovider.utils.**concept_adapter**(*obj*, *request*)
Adapter for rendering a skosprovider.skos.Concept to json.

**Parameters** **obj** (*skosprovider.skos.Concept*) – The concept to be rendered.

**Return type** dict

pyramid_skosprovider.utils.**label_adapter**(*obj*, *request*)
Adapter for rendering a skosprovider.skos.Label to json.

**Parameters** **obj** (*skosprovider.skos.Label*) – The label to be rendered.

**Return type** dict

pyramid_skosprovider.utils.**note_adapter**(*obj*, *request*)
Adapter for rendering a skosprovider.skos.Note to json.

**Parameters** **obj** (*skosprovider.skos.Note*) – The note to be rendered.

**Return type** dict

pyramid_skosprovider.utils.**parse_range_header**(*range*)
Parse a range header as used by the dojo Json Rest store.

**Parameters** **range** (*str*) – The content of the range header to be parsed. eg. *items=0-9*

**Returns** A dict with keys start, finish and number or *False* if the range is invalid.

CHAPTER 5

# History

## 5.1 0.5.0 (2014-12-19)

- Conceptschemes expose information on the subject they're tagged with. [BartSaelen]
- A new search endpoint for searching across conceptschemes was added. Search syntax is the same as for searching within a single scheme, but the collection parameter is not accepted. Two extra parameters were added for limiting the search to a subset of available conceptschemes. (#8)
- A new endpoint for looking up a certain URI was added. This endpoint does not redirect to an external URI, but lets a client know where more information about this URI can be found (eg. in which conceptscheme a concept lives). (#7)

## 5.2 0.4.0 (2014-10-23)

- Compatibility with skosprovider 0.4.0
- Drop support for Python 2.6 and Python 3.2.
- Expose notes on collections.
- Expose matches on concepts (collections don't have matches).
- Expose subordinate_arrays on concepts and superordinates on collections.
- Integrate concept scheme information. Concepts and collections passed through the service now contain the uri of the concept scheme they belong to. The concept scheme endpoint now also exposes information like a uri, a list of labels and notes.

## 5.3 0.3.0 (2014-06-24)

- Expose information about top concepts.
- Expose information about display top and display children.
- Fix a bug with returning concepts and collections not on the first page of data through the Range header. (#3)
- Added support for sorting. (#4, #5) [cedrikv]

## 5.4 0.2.0 (2014-05-14)

- Compatibility with skosprovider 0.3.0
- Added service documentation (#1)

## 5.5 0.1.1 (2014-04-10)

- Code coverage by coveralls.
- Removed unit tests from resulting package.
- Moved documentation to Sphinx.
- Reorganisation of tests.
- Changed to py.test as testrunner.
- Some Flake8 fixes.

## 5.6 0.1.0 (2013-05-16)

- Initial version
- Includes json views based on the interfaces skosprovider offers.
- Adds a skosprovider registry to the pyramid request.

CHAPTER 6

# Indices and tables

- *genindex*
- *modindex*
- *search*

## /c



## /conceptschemes



## /uris

## p

## C



## G



## L



## N



## P