SKOS Suggester Documentation

Release 0.0.1

Mobile Oxford team, IT Services, University of Oxford

May 06, 2016

Contents

CHAPTER 1

HTTP API

1.1 Get endpoint

Endpoint for retrieving SKOS concepts by their unique identifier.

GET /get

Get SKOS concepts by their URI

Example request:

```
GET /get?uri=http://id.worldcat.org/fast/869764 HTTP/1.1
Host: 127.0.0.1
Accept: application/json
```

Example request with multiple URIs:

```
GET /get?uri=http://id.worldcat.org/fast/1902995&uri=http://id.worldcat.org/fast/869764
Host: 127.0.0.1
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "_links":
        {"self":
            {"href":"/search?uri=http://id.worldcat.org/fast/1902995&uri=http://id.worldcat.org/
    "_embedded":
        {"concepts":
            [{"uri":"http://id.worldcat.org/fast/1902995",
            "prefLabel":"Depressions in motion pictures",
            "altLabels":[],"related":[]},
            {"uri":"http://id.worldcat.org/fast/869764",
            "prefLabel":"Commodore 64 (Computer)",
            "altLabels":[],
            "related":[{"label":"Electronic digital computers","uri":"http://id.worldcat.org/fas
]}}
```

Expect the same response if only one concept is requested (i.e. a list of one concept).

Query Parameters

- **uri** – unique identifier of a SKOS concept

Status Codes

- 200 OK – request done
- 400 Bad Request – Bad request (if you don't pass the parameter)
- 500 Internal Server Error – an exception occured

1.2 Search endpoint

Endpoint for searching SKOS concepts

GET /search
Search for SKOS concepts by labels

Example request:

```
GET /search?q=http HTTP/1.1
Host: 127.0.0.1
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
{"concepts":[
    {"uri":"http://id.worldcat.org/fast/950001",
    "prefLabel":"HTTP (Computer network protocol)",
    "altLabels":["HyperText Transfer Protocol (Computer network protocol)"],
    "related":[
        {"label":"Computer network protocols",
        "uri":"http://id.worldcat.org/fast/872279"}
    ]},
    {"uri":"http://id.worldcat.org/fast/878049",
    "prefLabel":"Cookies (Computer science)",
    "altLabels":["Persistent cookies (Computer science)","HTTP cookies (Computer science)","Magi
    "related":[
        {"label":"Data structures (Computer science)",
        "uri":"http://id.worldcat.org/fast/887978"}]}
    ]
}
```

Query Parameters

- **q** – search query
- **count** – number of results
- **page** – page (pagination)

Status Codes

- 200 OK – request done
- 400 Bad Request – Bad request (if you don't pass the parameter)
- 404 Not Found – Not found
- 500 Internal Server Error – an exception occured

1.3 Suggest endpoint

Endpoint for suggesting SKOS concepts

Mainly used for type-ahead

GET /suggest
Suggest SKOS concepts

Example request:

```
GET /suggest?q=secu HTTP/1.1
Host: 127.0.0.1
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{"concepts":[
    {"uri":"http://id.worldcat.org/fast/872484","prefLabel":"Computer security","altLabels":[],"
    {"uri":"http://id.worldcat.org/fast/869774","prefLabel":"Common Data Security Architecture (
    {"uri":"http://id.worldcat.org/fast/1005142","prefLabel":"Macintosh (Computer)--Security mea
    {"uri":"http://id.worldcat.org/fast/872590","prefLabel":"Computer software--Security measure
    {"uri":"http://id.worldcat.org/fast/872437","prefLabel":"Computer programs--Security measure
    {"uri":"http://id.worldcat.org/fast/872486","prefLabel":"Computer security--Computer program
    {"uri":"http://id.worldcat.org/fast/872485","prefLabel":"Computer security--Auditing","altLa
    {"uri":"http://id.worldcat.org/fast/925719","prefLabel":"Firewalls (Computer security)","alt
    {"uri":"http://id.worldcat.org/fast/872487","prefLabel":"Computer security--Costs","altLabel
    {"uri":"http://id.worldcat.org/fast/872488","prefLabel":"Computer security--Evaluation","alt
}
```

Query Parameters

- **q** – search query
- **count** – number of results
- **page** – page (pagination)

Status Codes

- 200 OK – request done
- 400 Bad Request – Bad request (if you don't pass the parameter)
- 404 Not Found – Not found
- 500 Internal Server Error – an exception occured

CHAPTER 2

Importers

2.1 RDF file importer

Imports SKOS concepts from an RDF file. This is the recommended method if you have a small dataset as it is easier to use.

2.1.1 Importing the file into SKOS suggester

Run the following command, where *[path to file]* is the path of the RDF file, and *[configuration file]* is the Dropwizard configuration file.

java -jar target/skos-suggester-1.0-SNAPSHOT.jar skosimport -f [path to file] [configuration file]

2.2 Jena TDB importer

Imports SKOS concepts from a Jena TDB dataset. This is the recommended method if you have a fairly large dataset (e.g. FAST (Faceted Application of Subject Terminology) topics).

2.2.1 Requirements

You need to install Jena to use the *tdbloader* command-line tool.

2.2.2 Importing RDF files into Jena TDB

Run the following command, where *[path to store]* is the path of a directory, and *[path to RDF file]* is the RDF file to be imported.

tdbloader2 –loc [path to store] [path to RDF file]

See TDB commands documentation for more information.

2.2.3 Importing TDB into SKOS suggester

Run the following command, where *[path to store]* is the path of the TDB directory, and *[configuration file]* is the Dropwizard configuration file.

java -jar target/skos-suggester-1.0-SNAPSHOT.jar tdbimport -d [path to store] [configuration file]

CHAPTER 3

Developers

3.1 Javadoc

3.1.1 uk.ac.ox.it.skossuggester

SkosSuggesterApplication

public class **`SkosSuggesterApplication`** extends Application<AppConfiguration>

Methods

initialize
public void **`initialize`**(Bootstrap<AppConfiguration> *bootstrap*)

main
public static void **`main`**(String[] *args*)

run
public void **`run`**(AppConfiguration *configuration*, Environment *environment*)

3.1.2 uk.ac.ox.it.skossuggester.cli

Skos

public class **`Skos`**
Utility methods used by importers RDF -> SolrDocument

Author martinfilliau

Fields

ALT_LABEL
public static final String **`ALT_LABEL`**

NS
public static final String **NS**

PREF_LABEL
public static final String **PREF_LABEL**

RELATED
public static final String **RELATED**

Methods

getDocument
protected static SolrInputDocument **getDocument**(Resource *res*)
Get a SolrInputDocument from a Resource

Parameters

- **res** – Resource to analyse

Returns SolrInputDocument to be ingested by Solr

SkosFileImporter

public class **SkosFileImporter** extends ConfiguredCommand<AppConfiguration>
Import SKOS concepts from an RDF file Easier to use than TdbImporter as it does not need an intermediary Jena TDB store, but might reach memory limits

Author martinfilliau

Constructors

SkosFileImporter
public **SkosFileImporter**()

Methods

configure
public void **configure**(Subparser *subparser*)

getDocsFromFile
protected Collection<SolrInputDocument> **getDocsFromFile**(File *file*, String *lang*)
Import a given RDF file to the search index

Parameters

- **file** – RDF file
- **lang** – RDF format (RDF/XML, N-TRIPLE, TURTLE or N3)

Throws

- **java.io.FileNotFoundException** –

Returns collection of SolrInputDocument

getDocsFromModel
protected Collection<SolrInputDocument> **getDocsFromModel**(Model *m*)
Import a Model

Parameters

- **m** – Jena Model

Returns collection of SolrInputDocument

run
protected void **run**(Bootstrap<AppConfiguration> *bootstrap*, Namespace *namespace*, AppConfiguration *configuration*)

TdbImporter

public class **TdbImporter** extends ConfiguredCommand<AppConfiguration>
Import SKOS concepts from a Jena TDB store More robust than SkosFileImporter for big data sets See http://jena.apache.org/documentation/tdb/

Author martinfilliau

Constructors

TdbImporter
public **TdbImporter**()

Methods

configure
public void **configure**(Subparser *subparser*)

run
protected void **run**(Bootstrap<AppConfiguration> *btstrp*, Namespace *namespace*, AppConfiguration *configuration*)

3.1.3 uk.ac.ox.it.skossuggester.configuration

AppConfiguration

public class **AppConfiguration** extends Configuration

Methods

getSolrLocation
public String **getSolrLocation**()

setSolrLocation
public void **setSolrLocation**(String *solrLocation*)

3.1.4 uk.ac.ox.it.skossuggester.dao

SkosConceptsDao

public class **SkosConceptsDao**
Encapsulates all queries to Solr

Author martinfilliau

Constructors

SkosConceptsDao
public **SkosConceptsDao** (SolrServer *solr*)
Constructor for SkosConceptsDao

Parameters

- **solr** – Instance of SolrServer (e.g. HttpSolrServer)

Methods

get
public Optional<SkosConcepts> **get** (List<String> *uris*)
Get documents by their unique IDs

Parameters

- **uris** – list of URIs

Returns SkosConcepts or absent

search
public Optional<SkosConcepts> **search** (String *query*, Integer *start*, Integer *count*)
Search for documents by a query string

Parameters

- **query** – string to search
- **start** – first document to retrieve
- **count** – number of documents to retrieve

Returns SkosConcepts or absent

suggest
public Optional<SkosConcepts> **suggest** (String *query*, Integer *start*, Integer *count*)
Search for documents by a query string Use the "suggest" handler which provides a light response

Parameters

- **query** – string to search
- **start** – first document to retrieve
- **count** – number of documents to retrieve

Returns SkosConcepts or absent

3.1.5 uk.ac.ox.it.skossuggester.health

SolrHealth

public class **SolrHealth** extends HealthCheck
Health check

Author martinfilliau

Constructors

SolrHealth
public **SolrHealth**(SolrServer *solr*)

Methods

check
protected Result **check**()

3.1.6 uk.ac.ox.it.skossuggester.jerseyutils

JsonIllegalArgumentExceptionMapper

public class **JsonIllegalArgumentExceptionMapper** implements ExceptionMapper<IllegalArgumentException>

Methods

toResponse
public Response **toResponse**(IllegalArgumentException *error*)

JsonIllegalArgumentExceptionMapper.ErrorMessage

public class **ErrorMessage**

Constructors

ErrorMessage
public **ErrorMessage**(String *message*)

Methods

getMessage
public String **getMessage**()

3.1.7 uk.ac.ox.it.skossuggester.representations

Related

public class **Related**
Represents a skos:related concept

Constructors

Related
public **Related**()

Related
public **Related**(String *label*, String *uri*)

Methods

equals
public boolean **equals**(Object *obj*)

getLabel
public String **getLabel**()

getUri
public String **getUri**()

hashCode
public int **hashCode**()

setLabel
public void **setLabel**(String *label*)

setUri
public void **setUri**(String *uri*)

toString
public String **toString**()

SkosConcept

public class **SkosConcept**
Represents a skos:concept

Author martinfilliau

Constructors

SkosConcept
public **SkosConcept**()

Methods

addAltLabel
public void **addAltLabel**(String *label*)
Add an alternative label to the concept

Parameters

- **label** – String

addRelated
public void **addRelated**(Related *related*)
Add a Related concept to the concept

Parameters

- **related** – Related

equals
public boolean **equals**(Object *obj*)

fromSolr
public static SkosConcept **fromSolr**(SolrDocument *doc*)
Get a SkosConcept from a SolrDocument

Parameters

- **doc** – SolrDocument

Returns SkosConcept

getAltLabels
public List<String> **getAltLabels**()

getPrefLabel
public String **getPrefLabel**()

getRelated
public List<Related> **getRelated**()

getUri
public String **getUri**()

hashCode
public int **hashCode**()

setAltLabels
public void **setAltLabels** (List<String> *altLabels*)

setPrefLabel
public void **setPrefLabel** (String *prefLabel*)

setRelated
public void **setRelated** (List<Related> *related*)

setUri
public void **setUri** (String *uri*)

toString
public String **toString** ()

SkosConcepts

public class **SkosConcepts**
Represents a list of SkosConcept

Author martinfilliau

Constructors

SkosConcepts
public **SkosConcepts** ()

SkosConcepts
public **SkosConcepts** (List<SkosConcept> *concepts*)

Methods

addConcept
public void **addConcept** (SkosConcept *concept*)
Add a SkosConcept to the list of concepts

Parameters

- **concept** – SkosConcept

equals
public boolean **equals** (Object *obj*)

fromSolr
public static SkosConcepts **fromSolr**(SolrDocumentList *docs*)
Get SkosConcepts from a SolrDocumentList

Parameters

- **docs** – SolrDocumentList

Returns SkosConcepts

getConcepts
public List<SkosConcept> **getConcepts**()

hashCode
public int **hashCode**()

setConcepts
public void **setConcepts**(List<SkosConcept> *concepts*)

3.1.8 uk.ac.ox.it.skossuggester.representations.hal

HalLink

public class **HalLink**
Represents a HAL link

Author martinfilliau

Constructors

HalLink
public **HalLink**()

HalLink
public **HalLink**(String *value*)

Methods

getValue
public String **getValue**()

setValue
public void **setValue**(String *value*)

HalLinks

public class **HalLinks**
Represents a collection of HAL Link

Author martinfilliau

Methods

getSelf
public HalLink **getSelf**()

setSelf
public void **setSelf**(HalLink *self*)

HalRepresentation

public class **HalRepresentation**
Represents a HAL representation

Author martinfilliau

Constructors

HalRepresentation
public **HalRepresentation**()

Methods

getEmbedded
public SkosConcepts **getEmbedded**()

getLinks
public HalLinks **getLinks**()

setEmbedded
public void **setEmbedded**(SkosConcepts *embedded*)

setLinks
public void **setLinks**(HalLinks *links*)

setSelfLink
public void **setSelfLink**(HalLink *link*)

3.1.9 uk.ac.ox.it.skossuggester.resources

Get

public class **Get**

Constructors

Get
public **Get**(SkosConceptsDao *dao*)

Methods

get
public HalRepresentation **get** (List<String> *uris*)

PaginationUtils

public class **PaginationUtils**

Author martinfilliau

Methods

getFirstResult
public static int **getFirstResult** (int *page*, int *count*)
Get the first result for a given page and count

Parameters

- **page** – page number
- **count** – number of results

Returns first result of the page

Search

public class **Search**

Constructors

Search
public **Search** (SkosConceptsDao *dao*)

Methods

search
public HalRepresentation **search** (String *query*, IntParam *page*, IntParam *count*)

Suggest

public class **Suggest**

Constructors

Suggest
public **Suggest** (SkosConceptsDao *dao*)

Methods

suggest
public HalRepresentation **suggest** (String *query*, IntParam *page*, IntParam *count*)

CHAPTER 4

Indices and tables

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

/get

`GET /get`, 1

/search

`GET /search`, 2

/suggest

`GET /suggest`, 3

T

U