

SPARQL endpoint

Introduction

Vocabularies in RDF published through RVA's Sesame instance are accessible through SPARQL 1.1 Query endpoints. (For details about the SPARQL 1.1 Query Language, see <http://www.w3.org/TR/sparql11-query/>.)

Finding SPARQL endpoints

The easiest way to find out about the availability of a SPARQL endpoint for a particular vocabulary is to check the vocabulary's entry in RVA. If the entry has a version that contains a link "Show SPARQL Endpoint", then that version of the vocabulary has a SPARQL endpoint available, and clicking on the link will display the SPARQL endpoint's URL. You need only append the SPARQL Query to construct a complete URL that will return results. See the section "Constructing a URL" below for more details.

Working with JavaScript-based SPARQL editors and services

Cross-Origin Resource Sharing (<http://enable-cors.org/>) is enabled for SPARQL endpoints served from vocabs.ardc.edu.au. This means that endpoint URLs can be entered into JavaScript-based SPARQL editors and services such as YASGUI (<http://yasgui.triply.cc/>).

Constructing a URL

URLs that access the SPARQL endpoints are constructed from the following components in sequence. (The value in the **Example** column for vocabulary identifier is artificially chosen; it does not correspond to a vocabulary published at RVA.)

Component	Construction	Example
The RVA hostname and API prefix	<code>http://vocabs.ardc.edu.au/repository/api/sparql/</code> or <code>https://vocabs.ardc.edu.au/repository/api/sparql/</code>	<code>http://vocabs.ardc.edu.au/repository/api/sparql/</code>
The vocabulary identifier	Vocabulary owner, vocabulary title, version title, separated by underscores	<code>ands_nal-agricultural-thesaurus_nal-thesaurus-2015-edition</code>
The SPARQL Query	The text <code>?query=</code> followed by the SPARQL Query, encoded using percent encoding (https://en.wikipedia.org/wiki/Percent-encoding)	<code>?query=SELECT%20*%20{?s%20?p%20?o}</code>

Combining the values in the example column yields: `http://vocabs.ardc.edu.au/repository/api/sparql/ands_nal-agricultural-thesaurus_nal-thesaurus-2015-edition?query=SELECT%20*%20{?s%20?p%20?o}`

Notes:

- The components of the vocabulary identifier are turned into "slugs" (https://en.wikipedia.org/wiki/Semantic_URL#Slug) before inclusion in the URL. In practice, this means conversion to lower case, and the replacement of all non-alphanumeric data with hyphens. You do not have to guess what a slug will be; each vocabulary available through Sesame on RVA has, on its view page, a sample link to a SPARQL endpoint.
- A small number of vocabularies of particular significance have vocabulary identifiers that do not follow the pattern shown in the table above. For example, the ARDC-Curated version of the ANZSRC 2020 Fields of Research vocabulary has vocabulary identifier `anzsrc-2020-for`. Please check each vocabulary's view page in the portal to confirm its endpoint URL.

Special URLs for current versions

RVA supports the publication of multiple versions of a vocabulary; access to the SPARQL endpoints of the different versions is effected by using the different version titles in the vocabulary identifier components of the URL, as shown in the previous section. However, for published (i.e., not marked as "deprecated") vocabularies there is an additional "shortcut" provided to access the endpoints of the version which has been tagged as the "current" version in the portal. In addition to the endpoints available with the version title, the same endpoints are also available automatically with `current` as the version title. For example, if the version with version title `nal-thesaurus-2015-edition` is tagged as the current version, the following two URLs generate the same results:

- http://vocabs.ardc.edu.au/repository/api/sparql/ands_nal-agricultural-thesaurus_nal-thesaurus-2015-edition?query=SELECT%20*%20{?s%20?p%20?o}
- http://vocabs.ardc.edu.au/repository/api/sparql/ands_nal-agricultural-thesaurus_current?query=SELECT%20*%20{?s%20?p%20?o}

Getting triples with CONSTRUCT queries

If you use a `SELECT` query, you will get results in the SPARQL Results format. If you would rather get results that are RDF triples in one of the standard serialization formats – such as RDF/XML or Turtle – use a SPARQL `CONSTRUCT` query. The most basic case would be to request all triples in the repository, i.e. to do an "export" of the repository. To do this, use the query:

```
CONSTRUCT { ?s ?p ?o } WHERE { ?s ?p ?o }
```

As always, spaces must be percent encoded. However, in this particular case, all spaces can be omitted. Therefore, to send this query using the same repository used in the examples above, use this URL: http://vocabs.ardc.edu.au/repository/api/sparql/ands_nal-agricultural-thesaurus_nal-thesaurus-2015-edition?query=CONSTRUCT{?s?p?o}WHERE{?s?p?o}

By default, a `CONSTRUCT` query returns triples in RDF/XML format. To get triples in a different format, send an `Accept` HTTP header with the MIME type of the desired format.

There's a list of the supported formats in the first table at: <https://rdf4j.org/documentation/reference/rest-api/#content-types>. Pick the format you want from the left column; specify the value in the right column in the "Accept" header.

For example, to get RDF/JSON, use this HTTP header:

```
Accept: application/rdf+json
```