The Journal of Open Source Software

Dassie: a database of subject terms and hierarchies in the Library of Congress Subject Headings

Michael Hucka[1]

1 Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, USA

DOI: 00.00000/joss.00000

Software

- Review
- Repository
- Archive

Submitted: 00 January 0000
Published: 00 January 0000

Summary

Dassie was developed to solve a simple need: to provide a fast way to search and browse the terms in the Library of Congress Subject Headings (LCSH) (Policy and Standards Division 2016). We converted a portion of the LCSH linked data graph into a database that makes explicit the "is-a" relationships between LCSH terms. The result, Dassie (a loose acronym for *"**da**tabase of **s**ubject term**s** and h**ie**rarchies"*), allows programs to use normal MongoDB network API calls to search for LCSH terms and their relationships.

Dassie comes with a MongoDB database setup/control program (written in Bash) and a command-line program (written in Python). The Dassie database server can be run on a users' desktop compute or a networked computer. The command-line program is convenient for doing simple look-ups and also serves as an example of how to write a Python client program that accesses the database over the network. (The same could be implemented using any of the different MongoDB drivers available for other programming languages). The following is an example of using the `dassie` command-line program to trace paths from the term `sh2008002926` to the top-most terms:

```
# dassie -t sh2008002926
======================================================================
sh85118553: Science
  sh85076841: Life sciences
     sh85014203: Biology
        sh2003008355: Computational biology
           sh2008002926: Systems biology

sh00007934: Science
  sh85076841: Life sciences
     sh85014203: Biology
        sh2003008355: Computational biology
           sh2008002926: Systems biology
======================================================================
```

Database structure

The LCSH database in Dassie was generated by beginning with the RDF file for the LCSH linked data, then processing the RDF triples to extract the `broader` and `narrower` relationships between terms while simultaneously skipping all the children's subject identifiers (i.e., terms whose names begin with `sj`), computing some additional properties, and finally storing everything in a MongoDB database. Each entry in the database is indexed by its LCSH identifier (for example, `sh89003287`) and has a structure of the following form, where the value types are always either a string, a list of strings, an empty list, or the value `None`.

```
{
  "_id": "string",
  "label": "string",
  "alt_labels": [ "string", "string", ...],
  "note": "string",
  "broader": [ "id", "id", ...],
  "narrower": [ "id", "id", ...],
  "topmost": [ "id", "id", ...]
}
```

The meanings of the fields are as follows:

Field	Description	SKOS RDF component
`_id`	The term identifier	URI of the term
`label`	The preferred descriptive label for the term	`core#prefLabel`
`alt_labels`	One or more alternative descriptive labels	`core#altLabel`
`note`	Notes (from LCSH) about the term	`core#note`
`broader`	List of hypernyms of the term	`core#broader`
`narrower`	List of hyponyms of the term	`core#narrower`
`topmost`	List of topmost hyponyms of the term	(computed)

Most of the fields in a Dassie entry are taken directly from the LCSH database, except for the field `topmost`. That field is computed by following hypernyms from a given entry until terms are reached that have no values for `broader`. The `topmost` field holds a list of the unique topmost hypernyms computing this way. (Note that there may be more than one path from a given term to a topmost term, and thus for a given number of topmost terms N, running `dassie -t` may show more than N paths.)

Security

To prevent security risks that would come from having unrestricted network access to the database, the database requires the use of a user name and password; these are set at the time of first creating installing and configuring Dassie database using `dassie-server` (described in the next section). By default, `dassie` uses the operating system's keyring/keychain functionality to get the user name and password needed to access the Dassie database over the network so that you do not have to type them every time you call `dassie`. If no such credentials are found, it will query the user interactively for the user name and password, and then store them in the keyring/keychain so that it does not have to ask again in the future.

References

Policy, and Library of Congress Standards Division. 2016. "LC Linked Data Service: Authorities and Vocabularies (Library of Congress)." http://id.loc.gov/authorities/subjects.html. http://id.loc.gov/authorities/subjects.html.