Exercise 4: Operating on Linked Data

Jonas Länzlinger

March 19, 2024

Note: All documents and the code are included on my forked GitHub repository: exercise-4

Task 1 - Take part to a decentralized social network

1. Location of my pod: Jonas-Pod
2. WebID: WebID

Task 1.1 - Update your FOAF profile

For this task I have update my FOAF profile document accordingly. Visit my WebID to check.

Task 1.2 - Query the distributed social graph

For the queries I have created .sparql files. To test my queries, just run the following commands below in the directory where the .sparql files are located. (the third command is for running the navigational query without link traversal)

1. comunica-sparql https://wiser-solid-xi.interactions.ics.unisg.ch/Jonas/profile/card -f task_1_2_1.sparql
2. comunica-sparql-link-traversal https://wiser-solid-xi.interactions.ics.unisg.ch/Jonas/profile/card -f task_1_2_2.sparql
3. comunica-sparql https://wiser-solid-xi.interactions.ics.unisg.ch/Jonas/profile/card -f task_1_2_2.sparql

Task 1.2 Part 2 - Describe Queries and differences

The comunica-sparql command takes an entry point. From this entry point, the HTTP query gets executed and the result gets returned. The comunica-sparql-link-traversal command does the same thing but when a result contains a link, another HTTP request to this new link gets executed. This iterative behavior continues until no link returns any new links.

1. `foaf:knows`

   returns all persons that are directly connected to the subject (my card in this case)

2. `(foaf:knows|^foaf:knows)+`

   returns all persons that are either directly connected to the subject (my card in this case), or are indirectly connected to the subject (the connection may be as far away due to the +-symbol). That means I get a list of persons that either I know directly, or the persons I know directly know directly. The +-symbol matches a path that connects the subject and object of the path by one or more matches. The ^-symbol matches inverse path (object to subject).

3. `(foaf:name|foaf:firstName|foaf:givenName)+`

   this line in the navigational query is another condition that has to be met for a record to be returned to the result set. Because people used different foaf-attributes for storing the name, I try to catch different possibilities here.

Executing the navigational query with link traversal results in a social graph where everyone is transitively known by me (how has used one of the three ways above to specify the name).

Executing the navigational query without link traversal results in getting only my own name as a result because my card has the foaf:name attribute in it. Other people are not in the result because their card doesn't get queried without the link traversal and therefore the select can not check if their card meets the condition of having a foaf:name attribute.

Task 2 - Build an application for your Solid pod

I have implemented the missing parts accordingly. To test the JaCaMo application, simply run the gradle task.

If something doesn't work, please get in touch with me (jonas.laenzlinger@student.unisg.ch)