Exercise 5

Kai Schultz

March 18 2024

Task 1: Take part to a decentralised social network

Task 1.1: Update your FOAF Profile

The Url to my Card: /kai/card#me
I added my Name and my mail address and you and Jonas as people that I know.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix solid: <http://www.w3.org/ns/solid/terms#>.

<>
    a foaf:PersonalProfileDocument;
    foaf:maker <https://wiser-solid-xi.interactions.ics.unisg.ch/kai/
       profile/card#me>;
    foaf:primaryTopic <https://wiser-solid-xi.interactions.ics.unisg.ch/kai
       /profile/card#me>.

<https://wiser-solid-xi.interactions.ics.unisg.ch/kai/profile/card#me>

    solid:oidcIssuer <https://wiser-solid-xi.interactions.ics.unisg.ch/>;
    a foaf:Person;
    foaf:name "Kai Schultz";
    foaf:mbox "kai.schultz@student.unisg.ch" ;
    foaf:knows <https://wiser-solid-xi.interactions.ics.unisg.ch/danaivach/
       profile/card#me> ;
    foaf:knows <https://wiser-solid-xi.interactions.ics.unisg.ch/Jonas/
       profile/card#me> .
```

Task 1.2: Query the distributed social graph

Both queries can also be found as sparql files on Github: nonTraversal — Traversal.
Use the cli commands comunica-sparql or comunica-sparql-link-traversal

1. Query the people you know based on your FOAF profile

```
"PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?person
WHERE {
 <https://wiser-solid-xi.interactions.ics.unisg.ch/kai/profile/card#me>
    foaf:knows ?person.
}
```

The above sparql query searches through all triples on the specified Iri (my profile card). The specified triple that were looking for is: <MyWebID><foaf:knows><Any Person>. That is what we did in the above query. From this query we select all distinct people we get.

1. Query the names of all people interconnected in the distributed social graph

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?name
WHERE {
  <https://wiser-solid-xi.interactions.ics.unisg.ch/kai/profile/card#me> (
     foaf:knows|^foaf:knows)+ ?person.
  ?person (foaf:name|foaf:firstName|foaf:givenName|foaf:nickName)+ ?name.
}
```

In the above query we use the property paths to query through the graph. The Comunica Link Traversal engine enables us to go through the external nodes and that is how we can crawl through the entire social graph.
While the same syntax is still possible without the link traversal we will in our case not be able to profit from it, since we do not execute any http requests to the found persons. If there were transitive relationships on on graph, so e.g.:
a owl:sameas :b.
:b owl:sameas :c.
We could do:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT *
WHERE
{
  ?x owl:sameAs+ ?y
}
```

This is an exampele of how a navigational query could be used on a single graph. So all in all there is no difference in syntax between the two queries since they are both just sparql. The executed HTTP requests differ, since one cli command can traverse link and the other cant. So executing it with comunica-sparql will result in 1 HTTP request and executing it with comunica-sparql-link-traversal will result in querying the entire reachable graph.

Task 2: Build an application for your Solid pod

All the code for the task can be found on Github: /src/env/solid/Pod.java
Code should be executed as described in the Exercise description.