
Assignment 6: Distributed Knowledge Graphs (5pt)

Deadline: Dec 19, 2023; 23:59 CET

In this exercise, you will gain hands-on experience with distributed knowledge graphs and Solid¹. Solid is an initiative to re-decentralize the Web that uses Linked Data to decouple applications from data. In Solid, users store their data in personal data stores (known as *Pods*) — and then grant explicit access to their data to the various applications they use. This assignment is structured as follows:

- In Task 1, each member of the group creates a personal Solid pod and uses it to store their profiles and data, including the group they belong to;
- In Task 2, you will query a social graph distributed across your group members' Solid pods.
- In Task 3, you will create a Solid-based application to extract information about listed movies from the created Linked Data environment.

① Your Personal Solid Pod (1.5pt)

First, each member of your group needs to **create their Solid pod**. For this assignment², you will be using our research group's instance of a Solid Community Server. You can register using an email address and a password at <https://solid.interactions.ics.unisg.ch/idp/register/>. Give your pod a name that is related to you, and remember your credentials since you will **not** receive an email confirmation after creating your pod. After creating your pod, you will see information similar to the screenshot on the next page. This informs you about:

- The location of your new pod, e.g., <https://solid.interactions.ics.unisg.ch/my-name-pod/>
- Your WebID (e.g., <https://solid.interactions.ics.unisg.ch/my-name-pod/profile/card#me>), which is a unique identifier that has been created for you and is used to authenticate with Solid applications. The WebID can be accessed using an HTTP GET request, which yields your profile, described using the *Friend-of-a-Friend (FOAF)* ontology³.

WARNING: If you edit your profile and introduce syntax errors such that the profile is not a properly formatted document in the Turtle format anymore, you will lose access to your Pod. You can use a Turtle editor (e.g., <https://ci.mines-stetienne.fr/teaching/semweb/turtle.html>) to edit your profile and ensure that it is valid Turtle syntax.

Next, a member of your group should create a new Turtle⁴ document in their Solid pod and indicate the group members (represented using their WebIDs) using the FOAF ontology. Do this by using a Pod editor such as Penny⁵. To make the created profile readable by anyone, use the Penny application to create the Access Control List⁶ for the created Turtle document and set the read permission for everyone in this control list. You also need to set the write permission to be able to make further changes to your document.

¹<https://solidproject.org/about>

²<https://github.com/HSG-DS-HS23/Assignment6>

³<http://xmlns.com/foaf/0.1/>

⁴<https://www.w3.org/TR/turtle/>

⁵<https://penny.vincenttunru.com/>

⁶<https://solid.github.io/web-access-control-spec/>

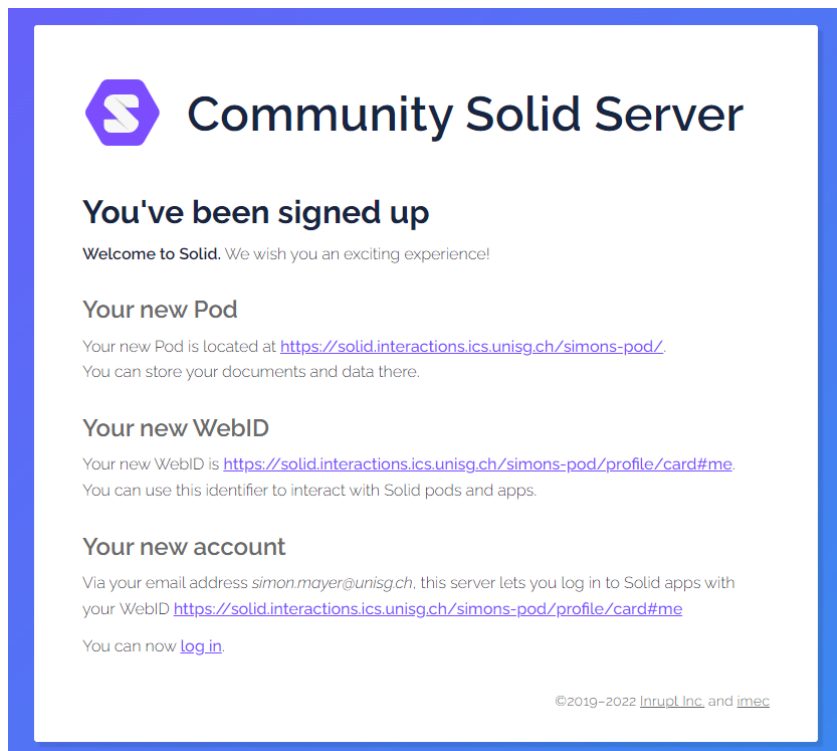


Figure 1: Sign-up Confirmation by the Solid Community Server.

Then, each group member should edit their FOAF profile to add triples that state (1) their name, (2) mailbox, and (3) the group they belong to. An example of a FOAF profile document can be found here: <https://solid.interactions.ics.unisg.ch/jlemee/profile/card#me>). Finally, use a Linked Data browser, such as these Chrome⁷ or Firefox⁸ plugins, to navigate the distributed social graph.

② Query the Distributed Social Graph (2pt)

Now that you have a distributed knowledge graph at hand, you can use SPARQL to query this graph. For instance, the following SPARQL query would retrieve J  r  my's full name from his FOAF profile:

```
1 SELECT DISTINCT ?name
2 WHERE
3 {
4 <https://solid.interactions.ics.unisg.ch/jlemee/profile/card#me> <http://xmlns.com/foaf/0.1/name> ?name.
5 }
```

To run SPARQL queries on Solid pods, we will use the Comunica⁹ knowledge graph querying framework — and you will need to install the following command-line tools:

- Comunica SPARQL, for querying RDF sources (e.g., documents, SPARQL endpoints): <https://www.npmjs.com/package/@comunica/query-sparql>. See also some documentation here: https://comunica.dev/docs/query/getting_started/query_cli/.
- Comunica SPARQL Link Traversal, for querying Linked Data on the Web: <https://www.npmjs.com/package/@comunica/query-sparql-link-traversal>

You can use Comunica to run SPARQL queries as follows, where you replace <POD-NAME> and <SPARQL-QUERY> with the actual Solid pod name and the query text you want to send:

⁷<https://chromewebstore.google.com/detail/openlink-structured-data/egdaiaihbdoiibopledjahjaihbmmjhdj>

⁸<https://addons.mozilla.org/en-GB/firefox/addon/rdf-browser/>

⁹<https://comunica.dev/>

```
1 comunica-sparql https://solid.interactions.ics.unisg.ch/<POD-NAME>/profile/card#me
  ``<SPARQL-QUERY>``
```

As an example, sending the above SPARQL query to the pod named `jlemee` should yield the response: `[{"name": "J          "}]` — Try it!

Your next task is to use your WebID as an **entry point for discovering other people** in the distributed social graph. Use `Comunica` to create and execute the following queries from the command line and note the results in your Report:

- 1.) Create a SPARQL query to retrieve the URI of the group you belong to from your FOAF profile. Note down this query in your Report.
- 2.) Query the names of all members of your group (using your profile as an entry point). To do this, you will need to use link traversal (command `comunica-sparql-link-traversal`). Try to run the query without link traversal as well, and see what happens. Note down both queries in your Report, together with the results that you obtain.

Note: By setting the logging level¹⁰ to debug (add `-l debug` at the end of each command), you can also observe the HTTP requests that are executed during querying.

③ A Solid-Based Application (1.5pt)

Use the application `MediaKraken`¹¹ to add movies to the Solid Pod of at least one member of your group. After that, the movies are included as files in the `movies` container of the pod but the `movies` folder is not public yet. Create an Access Control List for the added `movies` container using the Penny application, and set the read permission for everyone in this control list. Look at <https://solid.interactions.ics.unisg.ch/jlemee/movies/> for an example of such a created container.

Next, update the provided JavaScript project template (file `app.js` in folder **Task 3**) for creating a Node.js Web application which uses the `Comunica` framework¹², to retrieve the names and images¹³ of all movies stored in your Pod by `MediaKraken`. For this task, you merely need to complete the SPARQL queries present in the application.

Finally, we ask you to explore the possibilities of your Linked Data environment: Extend `app.js` with functionality that performs at least one additional SPARQL query to implement useful functionality for your users. For instance, you might query movies across all members of your team; or you might create a search engine that retrieves all movies that have been published in a given year.

Hand-in Instructions By the deadline, you should hand in a single **zip** file via Canvas upload. The name of this file should start with **a6** and contain the last names of all team members separated by underscores (e.g., `a6_lemee_jha_ciortea.zip`). It should contain the following files:

- All answers to the assignment questions in the given `REPORT.md`.
- Your code for **Task 3** as an archive if you do not provide a GitHub link in your Report.

Across all tasks in this and the other assignments in this course, you are **required to declare** any support that you received from others and, within reasonable bounds, any support tools¹⁴ that you were using while solving the assignment.

¹⁰<https://comunica.dev/docs/query/advanced/logging/>

¹¹<https://noeldemartin.github.io/media-kraken/>

¹²https://comunica.dev/docs/query/getting_started/query_app/

¹³Such as this: <https://image.tmbd.org/t/p/w500/fNOH9f1aA7XRTz11sA0x9iF553Q.jpg>

¹⁴It is not required that you declare that you were using a text-editing software with orthographic correction; it is however required to declare if you were using any non-standard tools such as generative machine learning models (e.g., GPT).