

Foundations of Semantic Web Technologies

Tutorial 11

Dörthe Arndt

WS 2022/23

Today, we will steal two entire exercises from Aidan Hogan (with his friendly permission). The idea is to do the exercise in class, so please bring your computer. We will build a proof-of-concept social network using the principles of Linked Data that will not be as shiny as Facebook, but will be *decentralised*, meaning that users keep control of their own data. The social network will be built with RDF using the Friend of a Friend vocabulary (<http://xmlns.com/foaf/spec/>). We will use RDF Explorer to interactively load and view the social network as a graph.¹

First we will play with some given examples.

- At <http://users.dcc.uchile.cl/~ahogan/foaf.ttl> you can find an example FOAF profile for Aidan (the hero of today's exercise), at <http://raw.githubusercontent.com/doerthe/swt/main/foaf.ttl> you can find mine. We can use RDF Explorer in order to load these data and visualise them as a graph. Visit the RDF Playground system at <http://rdfplayground.dcc.uchile.cl>, hit **BROWSE**, and enter one of the two given profile URLs, where it says "Insert an URL here". Click **SEARCH**.
- The graph will show a lonely central node. On the right-hand side, you will see a list of documents (currently we only have one). Click the checkbox for the one document shown. Now the graph for the full document is shown and the node is no longer lonely.
- Click the expand arrow to the left of the document in the list. Here you will see a list of properties mentioned in the RDF graph of the document. If you deselect a property, the graph will be drawn again without the triples with that property. If you deselect the document, all properties (and thus all the triples for that document) will disappear.
- Deselect the document and this time select only the properties `dc:creator`, `foaf:name`, and `foaf:knows`.
- Given a dereferenceable (Linked Data) node in the graph, we can double-click on it to retrieve data about it from the Web. (Not all nodes will be dereferenceable.) Double-click on one of the nodes for the two people that I know (an object of `foaf:knows`). You will see an option to **Extend Network**. Click the "START" button. On the left, a new document will appear. Repeat the same to dereference the other node of the person I know. Select both documents to show all of their triples. You can hover over an edge to see which document defines the corresponding triple.
- Next expand the node for my spirit animal. This will load RDF data for the animal (species) from DBpedia. Be advised that the next part *might* crash your browser. Select the entire document. While waiting, whisper words of gentle encouragement to your browser. The graph is large. Deselect this new document and this time add some properties that interest you one-by-one. Given that some documents are large, the system allows you to rather select properties one-by-one.²

¹Thanks to an extension of RDF Playground developed by Raúl Cid!

²We could alternatively use Wikidata rather than DBpedia for this lab, but the RDF documents it provides are *way* bigger.

Exercise 11.1. Time for you to join this social network. You will make a user profile in RDF about yourself (you can invent data if you prefer). We will describe these profiles in RDF (Turtle) using the FOAF vocabulary: `http://xmlns.com/foaf/spec/`. The profile will be published on the Web, so for personal data, feel free to make stuff up if you prefer.

- You can copy the example at `http://users.dcc.uchile.cl/~ahogan/foaf.ttl` into a text editor as a template. Note the different identifiers used for the document and for Aidan himself. Create your own FOAF profile (based on Aidan's), as follows:
 - First you will need to choose a location to publish your file online. You may publish it on your user account in Github³, or any other host you can use. Replace the `base` IRI in the Turtle file with the location that you have chosen: this should be the URL of the Turtle file once published online (keep the `http://` prefix; **better not to use a `https://` prefix, unless necessary**). If you're not sure about where to publish the file on the Web, you can postpone this step for now, but be sure to define the `base` IRI before publishing the document and moving onto Q2.
 - Add meta-data for the document of your profile, including the date, title, etc.
 - Choose a fragment identifier for yourself. The examples use the initials (`#ah`, `#da`) but you can/should change this (it appears three times). Note that `<>` and `<#ah>` are relative IRIs in Turtle appended to the base, so even if your initials are the same, don't worry: the complete IRI will still be unique.
 - Add your name(s), age, etc. (again, you can lie).
 - Add your image/avatar (`foaf:img`).
 - Add your homepage (if you have one; you can also use LinkedIn profile, Twitter page, etc.)
 - Say that you know Dörthe, and that you know the other colleagues in your class using their new IRIs (we can exchange that in class). Be sure to use the IRIs referring to them (the person, with a fragment identifier) and not their profile document.
 - Keep the first `foaf:interest` value (referring to the Semantic Web on DBpedia) and replace the others with your own interests from DBpedia (add at least two others besides the Semantic Web, but it can also be more). You can find DBpedia entities using the keyword search service available at `https://lookup.dbpedia.org/`. When adding these interests, you can use the prefix `dbr:...` or the full IRI starting with `<http://dbpedia.org/resource/...>` (it is the same IRI, just different Turtle syntax). Note that if the resource has special characters, like brackets or punctuation, you may need to use the full IRI.
 - Aidan created a small vocabulary for the lab `https://users.dcc.uchile.cl/~ahogan/social.ttl`. We will use the prefix `social:` for this henceforth.
 - Everyone has enemies. Add Aidan or Aidan Bot as your enemy.
 - Choose your own preferred values for the two properties `social:hasSpiritAnimal`, and `social:hasFavouritePlatonicSolid`. The value can come from DBpedia as in the previous step. Each property should take one value (the vocabulary defines the properties to be functional).
 - Add values about yourself for your choice of any two other properties from the FOAF vocabulary: `http://xmlns.com/foaf/spec/`. (The properties may or may not include `foaf:myersBriggs`, as you prefer.) In case the value of your property is an entity, not a string, try to find a good IRI for that value (e.g., using DBpedia) or create one if necessary.
 - Validate the syntax and visualise your profile with RDF Explorer using the local TEXT feature. Fix errors as needed.
 - Publish your FOAF profile in the location indicated by the base IRI you added earlier.

³If using Github, the location of your profile will be that of the *raw* file, not the HTML page previewing the file.

Exercise 11.2. Now its time to explore the graph of our social network. Load your profile into RDF Explorer using BROWSE and entering the URL where you published your profile. (Of course I won't control whether you browsed, but I encourage you to try. The original exercise expected screenshots, therefore the detailed instructions.)

- (a) Select the full document for your profile. Zoom the graph and move the nodes around so all of the data fit in the window.
- (b) Extend the network by dereferencing one of the nodes of at least one of the people you know (who we will henceforth refer to as your friend). Select the full document to display all of the data in the graph.
- (c) Extend the network further by dereferencing one of the nodes for the movies that your friend likes. Do not select the full document, but rather only the properties for its director and the people starring in it. (In case you see duplicate properties under the `dbo:` and `dbp:` prefixes, choose only the `dbo:` property; otherwise choose whatever is available.⁴ In case the properties are not available, try another movie.) Centre and zoom the graph on the data about the movie, and grab a screenshot.
- (d) Extend the network further by dereferencing the director node from the previous step. Select only the director property from the new document to see other movies that they have directed.
- (e) Refresh RDF Explorer and enter your FOAF profile again. For the available FOAF profiles, select only the properties `foaf:knows` and `social:hasEnemy`. Recursively expand the network with all of the non-DBpedia nodes that appear connected in the graph through these properties, selecting the same two properties in the new documents (where available) so they appear in the graph; keep going until you have successfully dereferenced at least ten documents, but if you like, you can keep going and dereference as many as you like. Some documents may not work due to an error in how the IRI is defined, or the document no longer being available; you can simply skip over these documents. Finally, in all of the FOAF profiles now collected, select the properties `foaf:name`, `foaf:interest`, `social:hasFavouritePlatonicSolid`, `social:hasSpiritAnimal` and `social:likesMovie`. Does anyone have shared values with you?

Note that the social network consists of RDF data. We could reason over it using RDFS/OWL, query over it using SPARQL, etc. The social network also connects to DBpedia, which connects to Wikidata, which connects to ... so it's more than just a social network: it's a small corner of the Web of Data.

⁴DBpedia defines two namespaces for properties. The `dbo:` properties are based on manual mappings of some common attributes in Wikipedia infobox templates, and tend to be cleaner. The `dbp:` properties are automatically extracted from unrecognised infobox attributes, and can *sometimes* be "dirtier". Hence `dbo:` is preferred, but often `dbp:` is fine.