Lab 1 – HTTP requests

1 Inspecting Wikidata

1. In your browser go to `wikidata.org`. By inspecting the network traffic when loading the page, can you figure out the `User-Agent` string used by your browser. Compare it to the one from other browsers on your computer or the ones obtained by other people on their computer. Can you figure out what the different parts of it mean?

2. What was the response code of the very first request to `wikidata.org`? What other requests did it lead to?

3. Can you find the first request that responded with a response code 200 (OK)? From the response header, where can you find the information about the language, format and encoding of the page.

4. Why, do you think, are there many other requests made afterwards?

5. While keeping an eye on the bottom of the list of requests in the browser's inspect window (network tab), start typing "University" in the search box of Wikidata. Did you see new requests appearing? What do you thing the ones starting with `api.php` are for? What is the format of the result (according to the response header)?

6. Can you find a request in the list using the `POST` method (`events?...`)? What do you think this one does? It sent data to the server. What was the format of that data according to the request header? You should be able to see the content sent through the request in a tab called "payload" (or something similar depending on your browser). Is it actually in the format specified by the request header? (if you don't see a POST request of this type, try to fill the login form and one should appear at the top of the list. However, this one does not specify a content type for the request).

7. Finish searching for "University of Lorraine" and click on the first result (`https://www.wikidata.org/wiki/Q4173330`). Now try going directly `https://www.wikidata.org/entity/Q4173330`. What happened?

8. You got redirected to an HTML resource. Can you figure out why from the request header?

9. Try to obtain the content of the same resource (`https://www.wikidata.org/entity/Q4173330`) using curl in verbose mode. What happens if you don't include the `-L` parameter? And if you do?

10. What is the format of the information obtained from the previous request (with `-L`). If not HTML, why?

11. Try to get the representation of the same resource (the University of Lorraine on Wikidata) in the HTML (`text/html`), Turtle (`text/turtle`) and N-Triples (`application/n-triples`) formats.

12. By removing the `-L` parameter and following re-directions manually, can you figure out what is the actual URL of the resource in the N-Triples format?

2 Following links in Wikidata

Using curl to retrieve RDF-based information and follow the links included, draw a graph of the relations that exist between the entity `https://www.wikidata.org/entity/Q4173330` (University of Lorraine), the entity `https://www.wikidata.org/entity/Q3214499` (LORIA) and all the people/places (towns, regions, countries) related to them.

3 Following links across sites

Starting from `https://dbpedia.org/resource/Nancy,_France` and using curl to get RDF representations and follow links using the relations `owl:sameAs` and `schema:sameAs`, list all the representations of the city of Nancy you can find outside of the `dbpedia.org` domain.

Note that some sites linked from an RDF representations might not provide an RDF representation for a resources and that not all forms of RDF representations can be found on all sites. For example, one site might provide resource representations in the RDF/XML format (`application/rdf+xml`), but not in the Turtle or N-Triples formats.