SW03

prince.prince

24 May 2023

1 Task 1:

Describe the connection between Linked Data and the Semantic Web.

The Semantic Web and Linked Data are two closely related concepts that together aim to make the web more intelligent, meaningful, and useful.The Semantic Web, is an extension of the current web that provides a standardized way of expressing relationships between data, so that it can be better understood and used by computers. On the other hand, linked data is a specific method for publishing structure data in a way it is more valuable and useful. The idea is to connect related data across different data sources on the web, creating a kind of global database. Linked data refers to data published on web in such a way that it is machine-readable, its meaning is explicitly defined, it is linked to other external data sets and can in turn be linked to form external data sets. LD uses URIs to identify things, get back data when we look up these resources, link the data using relations. Linked data is one of the invisible hand used by semantic web to reach its full potential.

2 Task 2:

Html code:

```
    <html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:foaf="http://xmlns.com/foaf/0.1/"
      xmlns:rdfa="http://www.w3.org/ns/rdfa#">
<head>
    <title>Prince's Interests</title>
</head>
<body>
    <div about="#me" typeof="foaf:Person">
        <h2 property="foaf:name">Prince Prince</h2>
        <p>
        Hello, my name is <span property="foaf:nick">Prince</span>,
    Email: <a href="prince.prince@uni-bielefeld.de">
        my mail:prince.prince@uni-bielefeld.de </a>,
```

```
    Phone: <a href="tel:+4915730489598">+49 15730489598</a>,
        and I am interested in
        <span rel="foaf:interest"
        resource="http://dbpedia.org/page/Coding">coding</span>,
        <span rel="foaf:interest"
        resource="http://dbpedia.org/page/Python">Python</span>, and
        <span rel="foaf:interest"
        resource="http://dbpedia.org/page/Quantum_computing">quantum computing</span>.

        </p>
    </div>
</body>
</html>
```

Figure 1: output of html file in browser

After converting RDFa into RDF using easyrdf, I got the output:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<http://njh.me/#me>
  a foaf:Person ;
  foaf:name "Prince Prince" ;
  foaf:nick "Prince" ;
  foaf:interest <http://dbpedia.org/page/Coding>,
  <http://dbpedia.org/page/Python>,
  <http://dbpedia.org/page/Quantum_computing> .
```

Now using the online tool RDF Grapher rdf-grapherto generate visualizations for RDF Turtle statements.

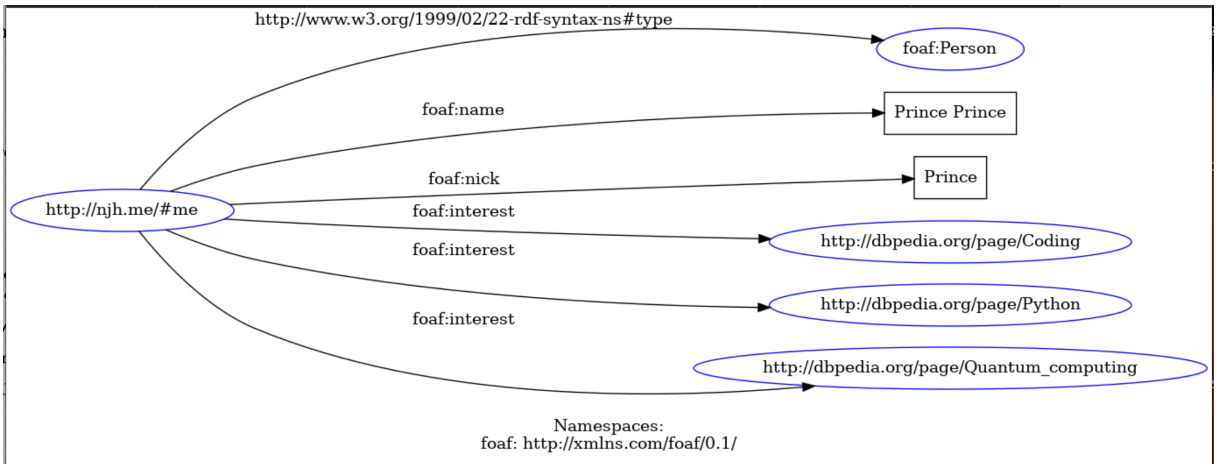

Figure 2: rdf graph for rdf turtle statements

3 Task 3:

Best Practices for Publishing Linked Data:

1. using URIs as name for things. Example from Task 2: The me in about="me" is a URI fragment, which is a type of URI.

2. using HTTPs URIs so that people can look up those names. Example from Task 2: The resources `http://dbpedia.org/page/Coding`, `http://dbpedia.org/page/Python`, and `http://dbpedia.org/page/Quantum_computing` are HTTP URIs that can be referenced to get information about these topics.

3. When someone looks up a URI, providing useful information. Example from Task 2: The HTML page includes information about "Prince Prince", me and my interests, and it links to DBpedia URIs where additional information can be found.

4. Including links to other URIs, so that they can discover more useful things. Example from Task 2: The rel="foaf:interest", this establish links between "Prince Prince" and my interests, which are represented by DBpedia URIs.

5. using standard data formats and vocabularies. Example from Task 2: The HTML page created by me uses the RDFa data format and the FOAF vocabulary, which are both widely recognized standards.

These are basic principles we can use for best practicing for publishing linked data, there are other things we can use to for data cleaning, data formatting and using SPARQL to query the data.

4 Task 4:

Results of the terminal step-by-step can be seen in below snippet from terminal:

Explanation of results: RDF publishers often use 303 redirects to distinguish between URLs for Web documents and URIs for Semantic Web resources. The idea is that when I fetch the URI of a non-document thing (e.g. a person or country), then the response will send me to the location of a document describing the thing.

In the first three commands 4, 5 and 6 , there is the 303 status code, indicates that the resource I requested can be found at a different URI, specified in the Location: header of the response, for '303 see other' means server is redirecting me to another URL where I can find the RDF/XML representation of the resource. In 5 and 6, it is happening multiple times, but in the end we reach the RDF/XML representation of the resource and get response status 200 OK.

To follow the redirect and retrieve the headers from the new location, I used the -L (or –location) option, you can observe it in 7, 8 and 9. This command tells curl to follow the redirect and fetch the headers from the new location. As the server is able to provide the data in RDF/XML format, we see a 200 OK response and a Content-Type: application/rdf+xml header. Otherwise, we might see another 303 See Other response if the server wants us to try a different format.

```
(base) prince@Princes-MBP ~ % curl http://example.com/
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Hel
vetica, Arial, sans-serif;

    }
    div {
        width: 600px;
        margin: 5em auto;
        padding: 2em;
        background-color: #fdfdff;
```

Figure 3: To test the cURL on my system, I got the HTML source code of the example web page.

```
(base) prince@Princes-MBP ~ % curl -I http://dbpedia.org/page/Bielefeld_University
HTTP/1.1 303 See Other
Server: nginx/1.18.0
Date: Tue, 23 May 2023 16:10:28 GMT
Content-Type: text/html
Content-Length: 153
Connection: keep-alive
Location: https://dbpedia.org/page/Bielefeld_University
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: HEAD, GET, POST, OPTIONS
Access-Control-Allow-Headers: Depth,DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,C
ache-Control,Content-Type,Accept-Encoding
```

Figure 4: 1st command run

```
(base) prince@Princes-MBP ~ % curl -I http://dbpedia.org/resource/Bielefeld_University
HTTP/1.1 303 See Other
Server: nginx/1.18.0
Date: Tue, 23 May 2023 17:12:04 GMT
Content-Type: text/html
Content-Length: 153
Connection: keep-alive
Location: https://dbpedia.org/resource/Bielefeld_University
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: HEAD, GET, POST, OPTIONS
Access-Control-Allow-Headers: Depth,DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,C
ache-Control,Content-Type,Accept-Encoding
```

Figure 5: 2nd command run

```
(base) prince@Princes-MBP ~ % curl -I -H "Accept: application/rdf+xml" http://dbpedia.org/resource/ Bielefeld_Uni
versity
curl: (6) Could not resolve host: application
HTTP/1.1 303 See Other
Server: nginx/1.18.0
Date: Tue, 23 May 2023 17:09:02 GMT
Content-Type: text/html
Content-Length: 153
Connection: keep-alive
Location: https://dbpedia.org/resource/
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: HEAD, GET, POST, OPTIONS
Access-Control-Allow-Headers: Depth,DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,C
ache-Control,Content-Type,Accept-Encoding
```

Figure 6: 3rd command run

```
(base) prince@Princes-MBP ~ % curl -I -L http://dbpedia.org/page/Bielefeld_University

HTTP/1.1 303 See Other
Server: nginx/1.18.0
Date: Tue, 23 May 2023 17:05:00 GMT
Content-Type: text/html
Content-Length: 153
Connection: keep-alive
Location: https://dbpedia.org/page/Bielefeld_University
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: HEAD, GET, POST, OPTIONS
Access-Control-Allow-Headers: Depth,DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,C
ache-Control,Content-Type,Accept-Encoding

HTTP/1.1 200 OK
Date: Tue, 23 May 2023 17:05:01 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 151499
Connection: keep-alive
Vary: Accept-Encoding
Server: Virtuoso/08.03.3328 (Linux) x86_64-generic-linux-glibc25  VDB
Expires: Tue, 30 May 2023 17:05:01 GMT
Link: <http://creativecommons.org/licenses/by-sa/3.0/>; rel="license", <http://dbpedia.org/data/Bielefeld_Univers
ity.rdf>; rel="alternate"; type="application/rdf+xml"; title="Structured Descriptor Document (RDF/XML format)", <
```

Figure 7: 1st command run with -L

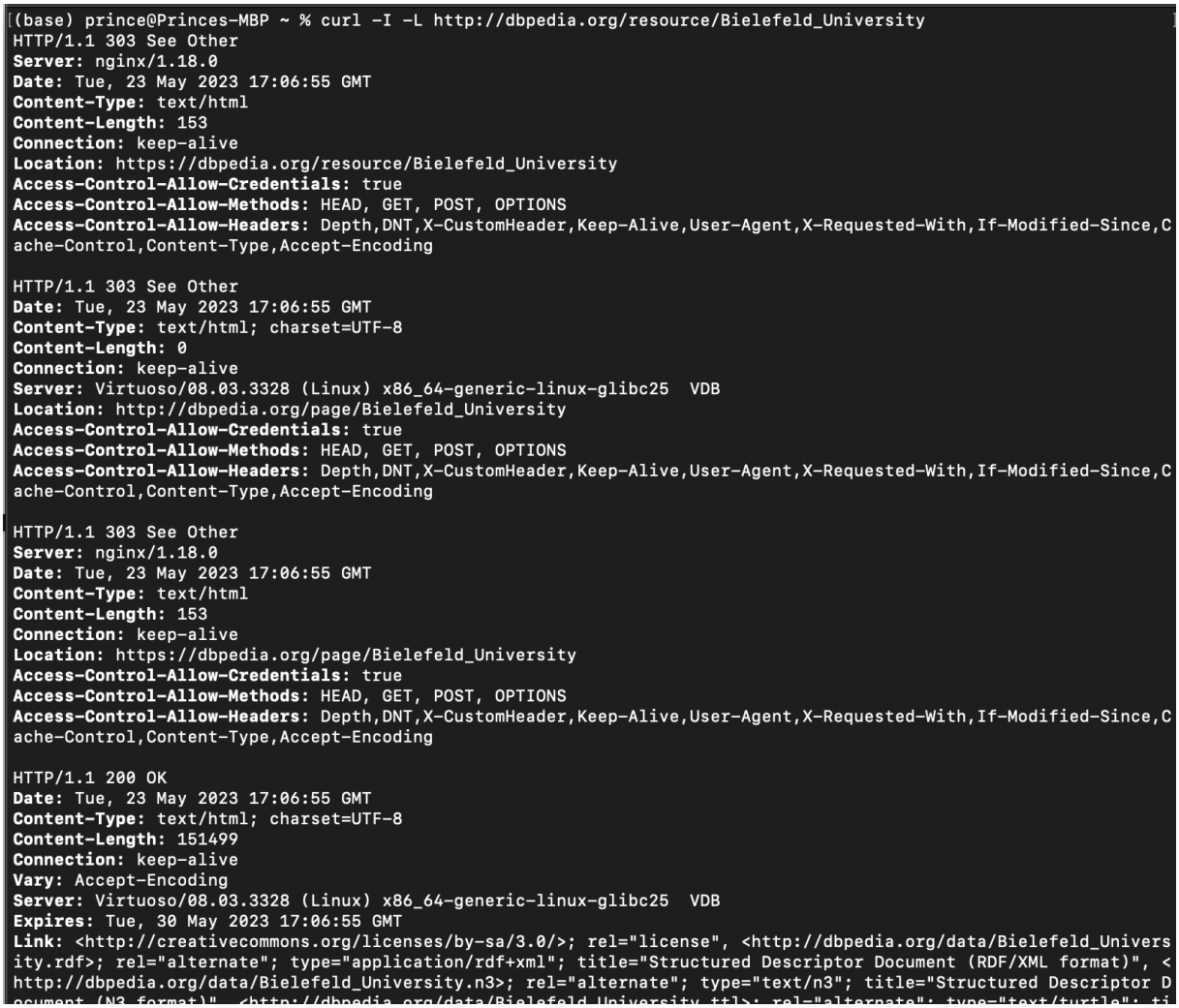

Figure 8: 2nd command run with -L

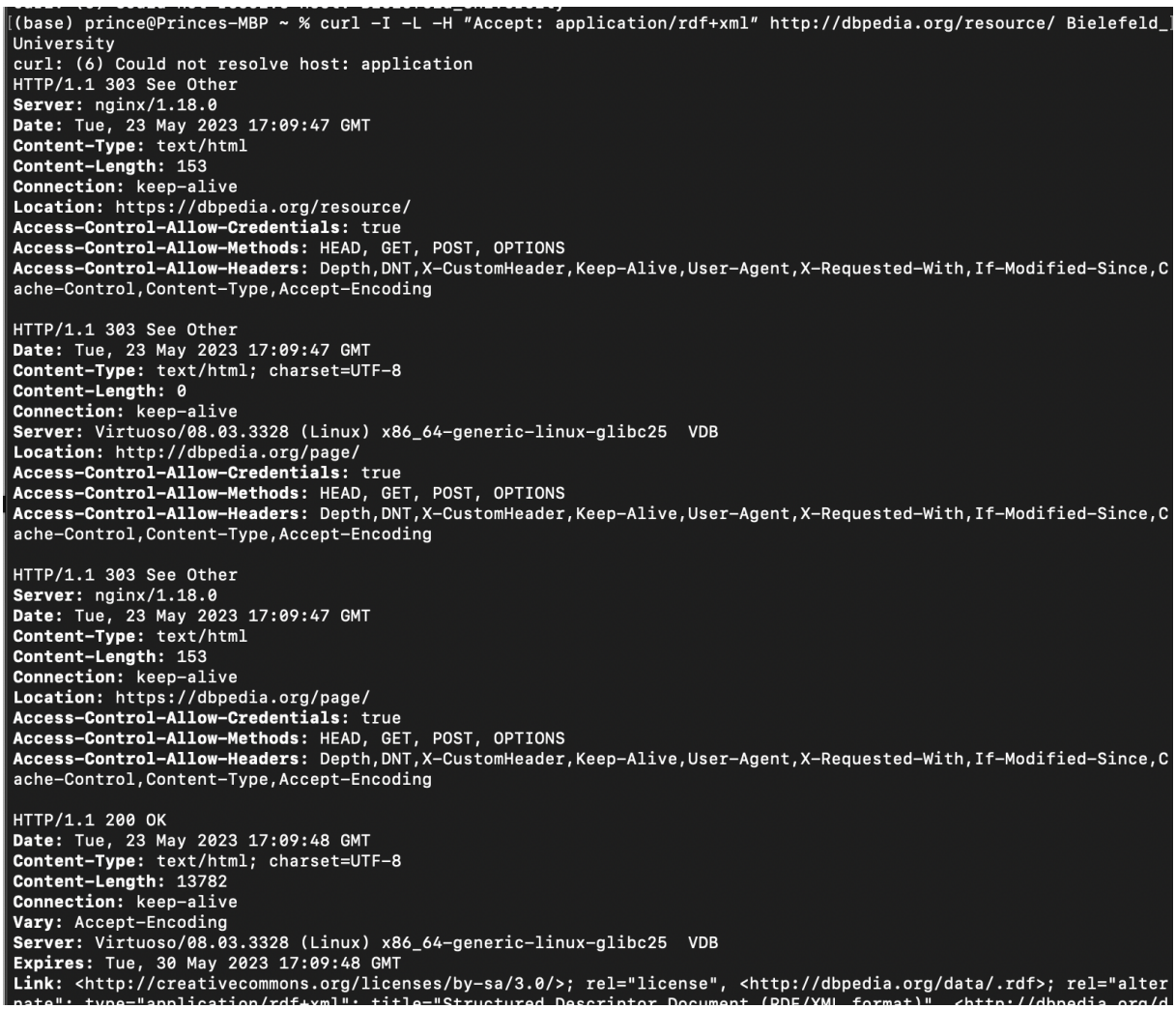

Figure 9: 3rd command run with -L