Expanding on the Norwegian Linked Open Dataset

Our system is created for deployment on a Linux Server, so all build instructions will assume a Debian-based Linux distribution. The guide is based on explaining the contents of each folder in the repository.

Table of Contents

source

The folder source, contains python scripts for extracting data from our sources and transforming the data into RDF/XML, as well as an Apache Airflow DAG.

Extract and Transform

To execute the scripts, we recommend creating a virtual environment.

Create and activate a new virtual environment `my-venv`:

```
$ python3 -m venv my-venv
```

```
$ source my-venv/bin/activate
```

To install the dependencies required to execute the Python scripts:

```
$ pip3 install -r requirements.txt
```

The JSON files that we fetch from Statens vegvesen are not complete, so we need to run a script that fetches all the information that we need and creates one large JSON file. To generate the JSON file:

```
$ python3 data_extraction.py
```

To generate the RDF triples alongside our parking ontology, run this command:

```
$ python3 rdf_transform.py
```

Apache Airflow

By setting up the virtual environment, you have already installed Airflow.

To configure it and start it, please follow Apache Airflows quick start guide https://airflow.apache.org/docs/apache-airflow/stable/start/local.html

config

Our system relies on third party software for publishing the linked data set generated by the Python scripts. It is not strictly necessary to setup all of this software, as we have made the interface of most of them publicly available.

The interfaces are:

- LodView: http://norpark.ml
- Apache Jena Fuseki: http://data.norpark.ml
- YASGUI: http://query.norpark.ml
- Cloud View: http://graph.norpark.ml/lod-cloud.svg
- Raw RDF data: http://raw.norpark.ml

Apache Jena Fuseki

Apache Jena Fuseki serves as our reasoner and private/internal SPARQL endpoint. It can be downloaded from here: https://jena.apache.org/download/index.cgi

We have deployed this as a Web-Application using Apache Tomcat9. A simple guide can be found here: https://jena.apache.org/documentation/fuseki2/fuseki-webapp.html#fuseki-web-application

To install Tomcat9 run: `$ sudo apt install tomcat9`

A small setup guide can be found here: https://tomcat.apache.org/tomcat-9.0-doc/appdev/index.html

The config file of Apache Fuseki can be found in config/fuseki/config.ttl, and should be placed in FUSEKI_BASE on your system. Modify the config file so that the path to the RDF file is correct.

LodView

LodView serves as our html representation generator, and generates this dynamically using our internal SPARQL endpoint. The source code can be found here: https://github.com/LodLive/LodView

This application is also deployed using Tomcat, please follow the instructions found in their wiki: https://github.com/LodLive/LodView/wiki/how-to%3A-install-and-configure-%28fast-method%29

The config file found in config/LodView, should be placed in the lodview/WEB-INF folder found under tomcat9/webapps

Apache2 HTTP Configuration

We assume general knowledge Apache2 HTTP Server with the reader, and provide configuration files for each site we host. Fuseki and LodView are hosted locally using Tomcat, but in order host them publicly on the internet, we need to setup a reverse proxy in Apache2.

The other config files are just normal setup files for websites in Apache2, and will be discussed in the respective section of the interface.

graph

Holds the Lod-Cloud generated by us as an SVG. This shows how our dataset will look on http://lod-cloud.net, when the cloud is updated next.

This is built using Lod-Cloud-Draw found here: https://github.com/lod-cloud/lod-cloud-draw Follow the instructions found in the wiki, and ensure that the input config file is: graph/lod-data.json

query

Apache Jena Fuseki out of the box gives the user a bit too much freedom, so we used YASGUI to implement a public SPARQL endpoint interface. This had the added upside of increased functionality as well.

YASGUI is a JavaScript library, which requires no installation.

After setting up the Apache2 configuration for the web-application it should work out of the box