INM713 Semantic Web Technologies and Knowledge Graphs

Laboratory 1: Setting-up the Infrastructure

Ernesto Jiménez-Ruiz

Academic course: 2020-2021
Updated: January 20, 2021

Contents

1 Git Repositories

I have created for the module two *github* repositories, one in Python and another in Java, for the reference codes of the laboratory sessions: `https://github.com/city-knowledge-graphs`

If you don not have already a GitHub account, you can set up a free account by visiting `https://github.com/`. GitHub is a public commercial widely service owned by Microsoft. Gitlab is an alternative solution (`https://about.gitlab.com/`).

The underlying technology for distributed version control/collaboration is called *git*. Check if git is installed in your computer (*e.g.*, type in the command line `git --help`). You can download/install git from `https://git-scm.com/downloads`).

GitHub Desktop (`https://desktop.github.com/`) may be an easy solution to manage your git repositories, although it is only available for macOS and Windows users.

1.1 Cloning and updating a repository

`clone` is the command to clone and download a git repository. First select the folder where you would like to download the codes and then run the command:

- Java:
  `git clone https://github.com/city-knowledge-graphs/java.git`
- Python:
  `git clone https://github.com/city-knowledge-graphs/python.git`

To update the local copy with new remote changes (*e.g.*, codes of future laboratory sessions) use the command pull: `git pull` from the respective local repository folders.

2 Python libraries

We will use Python 3. Check if you have python installed in your computer with `python --version` or `python3 --version`. Otherwise install from `https://www.python.org/downloads/`

I use Eclipse as Python editor, but Spyder IDE (`https://www.spyder-ide.org/`) seems to be the choice for many Python developers.

Required libraries:

- requests: HTTP library.
  - Installation: `https://pypi.org/project/requests/`
- RDFlib: a library to manage RDF graphs.

  - Installation: `https://pypi.org/project/rdflib/`
  - Documentation: `https://rdflib.readthedocs.io/en/stable/`
  - Examples: `https://github.com/RDFLib/rdflib`

- SPARQLWrapper: a python wrapper around a SPARQL service.
  - Installation: `https://pypi.org/project/SPARQLWrapper/`
  - Documentation and examples: `https://github.com/RDFLib/sparqlwrapper`

- Owlready: a package for ontology-oriented programming in Python.
  - Installation: `https://pypi.org/project/Owlready2/`
  - Documentation: `https://owlready2.readthedocs.io/en/latest/intro.html`

3 Java libraries

The created github project relies on maven (`https://maven.apache.org/install.html`) to deal with the library dependencies. We will use Java 8 or higher `https://www.oracle.com/uk/java/technologies/javase-downloads.html`. I recommend Eclipse IDE as editor (`https://www.eclipse.org/eclipseide/`).

Required libraries:

- Apache Jena: This API will lead with the Ontology and RDF management, and the SPARQL processing.
  - Documentation: `https://jena.apache.org/getting_started/index.html`
  - Installation. Maven is recommended. Alternative options: `https://jena.apache.org/download/index.cgi`

4 Code overview

The codes provided in both Python and Java (folder/package *lab1*) give an overview about the functionalities we will use in the lab sessions. They include methods for:

- Loading and ontology and printing its classes (*loadOntology* script/class).
- Loading and querying a local RDF knowledge graph (*loadRDFGraph* script/class).
- Querying a remote RDF knowledge graph via a SPARQL endpoint (*queryEndpoint* script/class).
- Retrieving pre-computed knowledge graph embeddings (*getKGEmbeddings* script/class).

You should be able to run the python scripts or the java classes without compilation errors (this is the main target of this session). At this stage you do not need to understand all the details, but you can start giving a look to the codes and get familiar with the libraries.

5 Protégé ontology editor

Protégé is an editor of OWL ontologies (`https://protege.stanford.edu/`). Install Protégé (Desktop version), load the Pizza ontology (Open from URL: `https://protege.stanford.edu/ontologies/pizza/pizza.owl`) and explore the interface.

This is just the first contact with the editor, we will explore its capabilities once we know a bit more about ontologies and OWL.