2. Handling RDF in Java: The Jena Framework

The goals of this lab are to:

1. Set up Jena in Eclipse

2. Create and manipulate RDF graphs

3. Build a SPARQL endpoint using Jena Fuseki

4. Query RDF graphs.

1.1 Set Up Jena in Eclipse

Apache Jena (or Jena in short) is a free and open source Java framework for building Semantic Web and Linked Data applications. The framework is composed of different APIs interacting together to process RDF data.[1]

There are two ways to set up Jena in Eclipse: first, by including manually the Jena JAR library and its JAR dependencies into the project's build path;[2] and second, by using Apache Maven, a tool to help Java projects manage their dependencies on library code, to declare a dependency on the core of Jena in your project's `pom.xml` file so that you will get automatically the consistent set of library files that Jena depends on.[3]

Create a simple program to load the following triple into an object of the `Model` class of Jena to test your set-up:

```
<http://example.org/resource/$student-name$>
  <http://example.org/property/said> "Hello World!"@en .
```

where you replace `$student-name$` with a name of your choice.

1.2 Create and Manipulate RDF Graphs

For Lab 1 we assumed that you are given a project by the City of Bolzano to develop a Semantic Web backend for tourism and a Semantic Web application on top of it. The Semantic Web backend would include an RDF triple store, a SPARQL query server and an OWL reasoner engine.

In Lab 1, you already modeled tourism information in RDF, creating a data set about hotels, attractions, restaurants, etc.

---

[1]`https://jena.apache.org/`
[2]`http://www.iandickinson.me.uk/articles/jena-eclipse-helloworld/`
[3]`https://jena.apache.org/download/maven.html`

Now, in this lab your task is to load such RDF data into Jena, manipulate it, and output it. To start, you need a file with tourism data in RDF, preferrably in Turtle notation. You can use either the file you created in Lab 1 or the RDF document about Hotel Laurin that you can download from `http://bit.ly/1yGRmXk`.

What you need to do in your code is the following:

1. Load your RDF file into an object of the `Model` class of Jena.

2. Write Jena code that adds the RDF description of another hotel in Bolzano.

3. Use Jena to generate Turtle and RDF/XML documents about your extended data set.

1.3 Build a SPARQL Endpoint using Jena Fuseki

Fuseki is a SPARQL server. It provides REST-style SPARQL HTTP Update, SPARQL Query and SPARQL Update using the SPARQL protocol over HTTP.[4]

From your exported tourism data, you are asked to set up a SPARQL endpoint for the data using Fuseki. One way to do it is by downloading a Fuseki server from the Fuseki site, and then running the server using the command form:[5]

```
fuseki-server --file=FILE /DatasetPathName
```

The command will create an empty, in-memory (non-persistent) dataset and then load `FILE` into it. Fuseki uses port 3030 as its default port. You can access Fuseki on your own computer at `localhost:3030`.

1.4 Query RDF Graphs

Over your SPARQL endpoint, execute the following SPARQL queries and show the results:

1. Give all 4-star hotels

2. Give all hotels that facilitate free breakfast

3. Give the name and address of each hotel.

---

[4]`http://jena.apache.org/documentation/serving_data`
[5]Make sure your Java version is ≥ 1.7