Who am I?

Sergio Fernández

Partner Technology Manager at Redlink GmbH

also...

External Lecturer at Fachhochschule Salzburg
Member of The Apache Software Foundation

@wikier

http://linkedin.com/in/sergiofernandez

http://www.wikier.org
What is Apache Marmotta?

- An **Open Platform for Linked Data**
  
an open implementation of a Linked Data Platform that can be easily used, extended and deployed by organizations who want to publish Linked Data or build custom applications on Linked Data.

- Key features:
  - Read-Write Linked Data server
  - RDF triple store with transactions, versioning and rule-base reasoning
  - LDP, SPARQL and LDPath query
  - Transparent Linked Data Caching
  - Integrated basic security mechanisms

What is Linked Data?

- The Semantic Web is a **Web of Data**
- Semantic Web technologies (**RDF**, **OWL**, **SKOS**, **SPARQL**, etc.) provide an environment where applications can query that data, draw inferences using vocabularies, etc.
- **Linked Data** lies at the heart of what **Semantic Web** is all about: large scale integration of, and reasoning on, data on the Web.
- A typical case of a large Linked Dataset is **DBPedia**, which, essentially, makes the content of Wikipedia available as Linked Data.
What is RDF?

- The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model.

- RDF is **directed labeled graph**, where:
  - nodes are resources;
  - edges represent the named links between two resources;
  - the composition of one resource (subject) linked (with a predicate) to another (object) is known as "RDF triple";
  - a set of triples form a RDF graph.
Querying in Marmotta

Currently Marmotta provide three main means of querying:

- **LDP 1.0** (Linked Data Platform)
  - a W3C protocol based on HTTP for managing Linked Data resources
  - [http://www.w3.org/TR/ldp/](http://www.w3.org/TR/ldp/)

- **SPARQL 1.1** (SPARQL Protocol and RDF Query Language)
  - a W3C RDF query language and protocol
  - [https://www.w3.org/TR/sparql11-query/](https://www.w3.org/TR/sparql11-query/)

- **LDPath**
  - a path language for Linked Data
  - similar to XPath for XML
  - [http://marmotta.apache.org/ldpath/language](http://marmotta.apache.org/ldpath/language)
GeoSPARQL

- The **OGC GeoSPARQL** standard supports representing and querying geospatial data on the Semantic Web.
- GeoSPARQL defines a vocabulary for representing geospatial data in RDF, and a SPARQL extension for processing geospatial data.
- It makes use of both WKT (Well Known Text) and GML for representing geometries as literals.
There are three key classes in the GeoSPARQL ontology:

- **geo:SpatialObject**
  - a superclass of both Features and Geometries;

- **geo:Feature**
  - a thing that can have a spatial location; i.e., a park or a monument etc.;

- **geo:Geometry**
  - a representation of a spatial location; i.e., a set of coordinates.

Namespace: http://www.opengis.net/ont/geosparql#
GeoSPARQL basic data model
GeoSPARQL in Marmotta

- More precisely we should say "GeoSPARQL in KiWi"
  - KiWi is our triple store based on relational databases
  - Marmotta also supports many other Sesame-based triple stores as backend

- Support implemented based on PostGIS for PostgreSQL
  - Support not available for other databases

- All further technical details available at [https://wiki.apache.org/marmotta/GSoC/2015/MARMOTTA-584](https://wiki.apache.org/marmotta/GSoC/2015/MARMOTTA-584)
  Documentation at [http://marmotta.apache.org/kiwi/geosparql](http://marmotta.apache.org/kiwi/geosparql)
GeoSPARQL implementation approaches

Two approaches were mainly considered for implementing GeoSPARQL:

- **Materialization**
  - Pros: fast querying
  - Cons: materialization is computationally expensive, requires more storage capacity and native operators

- **Query translation**
  - Pros: direct comparison, optimal storage and no need of native operators
  - Cons: slow querying

In Marmotta we decided to go for the first one.
GeoSPARQL coverage

In 3.4.0 Marmotta will* support:

- Simple Features Topological Relations
- Egenhofer Topological Relations
- RCC8 Topological Relations
- Non-Topological Functions

(*) still under development at MARMOTTA-584 branch
GeoSPARQL example

Simple query to get all geometries that are contained by other.

```
SELECT DISTINCT ?label
WHERE {
    ?reg1 a geo:provincia ;
    rdfs:label "Madrid"@es ;
    geoes:hasExactGeometry ?geo1 .
    ?geo1 geo:asWKT ?wkt1 .

    ?reg2 a geo:municipio ;
    rdfs:label ?label ;
    geoes:hasExactGeometry ?geo2 .
    ?geo2 geo:asWKT ?wkt2 .

    FILTER (geof:sfContains(?wkt1, ?wkt2))
}
ORDER BY ?label
LIMIT 10
```

Particularly this example queries for the first ten municipalities in the region of Madrid.
let's demo!
questions?
we're HIRING!

- Java Engineer (Solr)
- PHP Web Developer
- Interns

http://redlink.co/careers
Kösz!
The work presented here has been developed in the context of the TourPack project, partially funded by the Austrian Research Promotion Agency (FFG) IKT der Zukunft program under grant agreement no. 845600.

Acknowledgements

Thanks to the student Francisco Xavier Sumba Toral for contributing the initial GeoSPARQL implementation as part of his project during the Google Summer of Code 2015.

Thanks to Google for such awesome open source program!