

Capabilities

Explore & Query is visual front-end to build and execute SPARQL queries.

It supports 2 type of bindings

1. HTTP binding, which means you can connect over public SPARQL endpoints to any RDF based Linked Data source
2. Jena binding to Virtuoso Universal server

What does it allow you to do

You can query any public SPARQL endpoint, thus instead of using their default SPARQL editors to type the query text in, you can Explore & Query using Vinge' visual front-end.

But if you use the Virtuoso triple store and have it installed at your premise, you can configure this software to connect to your database directly over Jena/JDBC and use to empower your staff to work with your own semantic data.

Installation

The software is provided as Tomcat web application and comes as a war file.

In order to install:

- Setup Apache Tomcat. It has been tested with v7 but also runs on v 6
 - Make sure in your server.xml file, the <Connector> you use has explicitly set URIEncoding="UTF-8"
- Download the **ExploreAndQuery.war** file and save it locally
- In Tomcat Manager App - WAR file to deploy section - select the war file and deploy

GraphViz

You need to install the GraphViz on the machine where Tomcat is running. Install latest available for your OS

Installation done

Verify installation

In browser (Chrome, Safari or Firefox) go to address **http://<host>:port/ExploreAndQuery** , where <host> and port are specific to your Tomcat installation

You should get to Enter page

Configuration

The web app has 4 configuration files, where you specify your stuff. Find them in your Tomcat installation structure under webroot/ExploreAndQuery.

WEB-INF/web.xml

you need to change at least two first <init-param> values
Others are optional

<param-name>rootURI</param-name>

This is the root URI of the web application.

example: <http://localhost:8080/ExploreAndQuery>

this is also default if you run the client locally

<param-name>GraphVizDOT</param-name>

This is the location of GraphViz executable - dot file

example Linux: /usr/bin/dot

example win: C:\Program Files (x86)\Graphviz2.30\bin\dot.exe

<param-name>logsDir</param-name>

this is the path to the location where application writes query logs. Create it somewhere on the server where Tomcat is running.

example: /vol/ebs/Audit/

NB! must have separator at the end

<param-name>odbc_conn</param-name>

Optional: If you only connect over HTTP sparql endpoint protocol, leave it empty

If you have Virtuoso server installed and what to connect to your data over JDBC then set the value to host:port :

example: localhost:1111

You also need to create predefined users in your Virtuoso system, whose password is “guest”

You also need to assign SPARQL_SELECT role to those users
Example; user1/guest , user2/guest

`<param-name >resultsDir</param-name>`

this is the path to the location where application writes all kind of results from queries and navigation. Create it somewhere on the server where Tomcat is running.

example: /vol/ebs/QueryResults/

NB! must have separator at the end

WEB-INF/log4j.lcf

In this file change file name where Tomcat writes the logs

example:

log4j.appender.R.File=/opt/Apache/apache-tomcat-7.0.37/logs/EQ.log

You need to reload the ExploreAndQuery application in Tomcat manager after changing these settings.

Servers.config

In this file you specify the locations of the servers to connect to.

Currently there are 2 preconfigured SPARQL endpoints, and their entries look like this:

http://dbpedia.org/sparql,SPARQL_Service,DBpedia public endpoint

http://enipedia.tudelft.nl/sparql,SPARQL_Service,Enipedia - Energy Industry public endpoint

Each entry must consist of comma separated parts:

- URI of the endpoint / JDBC connection location
- type of the service.3 options available
 - SPARQL_Service
 - SPARQL_Service/OWLIM
 - Virtuoso
- Name of server that appears to the user

You can add at any time new entries. In order them to pop up in selection list, you need to reload the web app in Tomcat manager

Prefixes.config

In order to show names of classes and properties without long URI's but using prefixes, you can setup prefix mapping in this file. This is always very source specific and it may take some effort to extract the namespaces, as there are sometimes many in same source.

Example of the file which is self-explanatory.

Must use one space between prefix and uri, no colon and no brackets. You can use comments with hash sign and empty lines.

```
#Enipedia
enip_d http://enipedia.tudelft.nl/wiki/
enip_m http://enipedia.tudelft.nl/wiki/Category:
enip_p http://enipedia.tudelft.nl/wiki/Property:
enip_w http://enipedia.tudelft.nl/wiki
```

```
#DBpedia
dbp_d http://dbpedia.org/resource/
dbp_m http://dbpedia.org/ontology/
dbp_p http://dbpedia.org/property/
```

Spotfire output

In the demo videos we show how to pass query results directly to Spotfire for initial quantification and additional filtering.

This feature is not enabled by default and it requires that you run Virtuoso database and you have access to Spotfire. There is 30-days Spotfire trial available downloadable from our [site](#) and there is Virtuoso Open Edition as well.

If you want to set this feature up please talk to us and we will make it available.