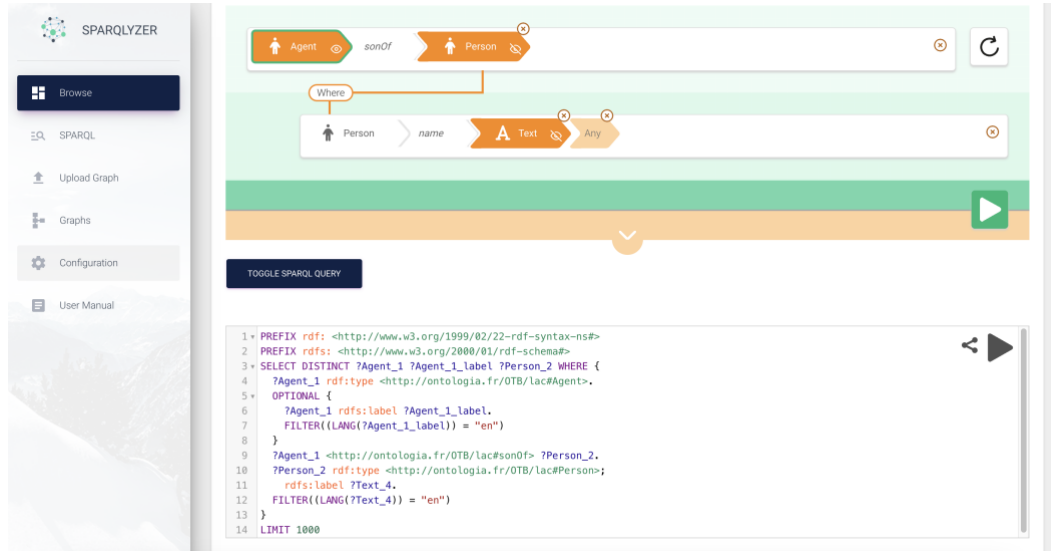


# User Manual: SPARQLyzer

## Description

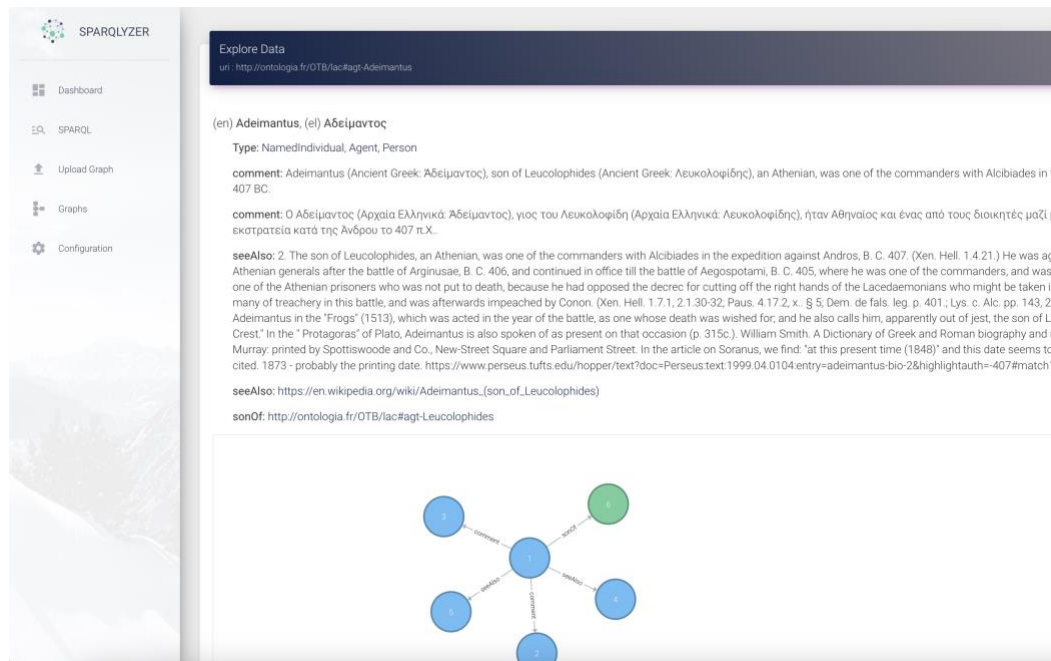


The screenshot displays the SPARQLyzer web application interface. On the left is a sidebar with navigation options: Browse, SPARQL, Upload Graph, Graphs, Configuration, and User Manual. The main area is divided into two sections. The top section is a graphical query builder with a 'Where' clause. It shows a path: Agent (sonOf) Person (name) Text (Any). Below this is a 'TOGGLE SPARQL QUERY' button. The bottom section is a code editor containing the following SPARQL query:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 SELECT DISTINCT ?Agent_1 ?Agent_1_label ?Person_2 WHERE {
4   ?Agent_1 rdf:type <http://ontology.fr/OTB/Lac#Agent>.
5   OPTIONAL {
6     ?Agent_1 rdfs:label ?Agent_1_label.
7     FILTER((LANG(?Agent_1_label)) = "en")
8   }
9   ?Agent_1 <http://ontology.fr/OTB/Lac#sonOf> ?Person_2.
10  ?Person_2 rdf:type <http://ontology.fr/OTB/Lac#Person>.
11  rdfs:label ?Text_4.
12  FILTER((LANG(?Text_4)) = "en")
13 }
14 LIMIT 1000
```

SPARQLyzer is standalone web application with a key role in allowing users to:

- Browse/explore data using the [sparnatural](#) library on uploaded graphs
- Upload RDF data (.owl, .Fl, .rdf) to a triplestore
- Query a SPARQL endpoint
- Delete graphs from the triplestore




The screenshot displays the 'Explore Data' view in SPARQLyzer. The top bar shows the URL: `http://ontology.fr/OTB/lac#agt-Adeimantus`. Below this, the entity is identified as (en) Adeimantus, (el) Αδείμαντος. The type is NamedIndividual, Agent, Person. A comment describes Adeimantus as an Athenian commander. A detailed comment in Greek provides historical context. A 'seeAlso' section lists related resources, including a Wikipedia entry and a Perseus text. The 'sonOf' property is linked to Leucolophides. At the bottom, a graph visualization shows the entity and its relationships with other nodes.

The tool has the ability to **dynamically** export all classes and properties of the uploaded data, allow the users to construct SPARQL queries in a graphical and easy to use interface, execute the queries and start exploring/navigating the data.

Technical README can be found [here](#).

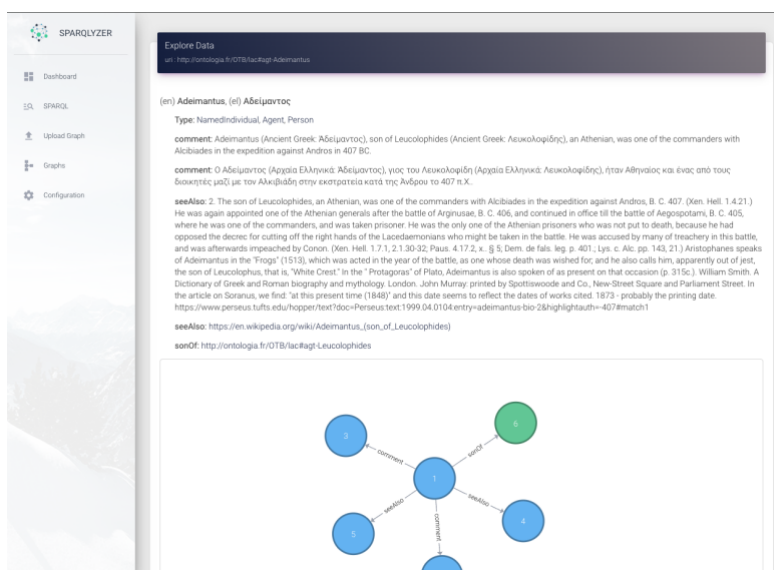
## Quick Start

The following steps must be taken for users to upload RDF data and start exploring:

1. Navigate to the **Upload Graph** tab and fill all in the fields accordingly [more details here]:
  - Graph URI: the URI of the uploaded ontology **as it appears in the data** (eg. For the LACRIMALit ontology the graph URI should be <http://ontologia.fr/OTB/lac>).
  - Triplestore username and password (credentials will be sent to eligible users)
  - Upload the RDF data from your computer by clicking the  button.
  - Click on the Upload button. If all the above fields are filled in correctly, the graph will be ready to start querying.
2. Navigate to the **Browse** tab and select the desired graph from the drop-down list. The tool will dynamically extract all the classes, properties and domains of the selected ontology display them in the sparnatural environment and construct SPARQL queries in a graphical user interface.
3. After constructing the desired query, it can be executed by clicking on the button below [more details [here](#)]:

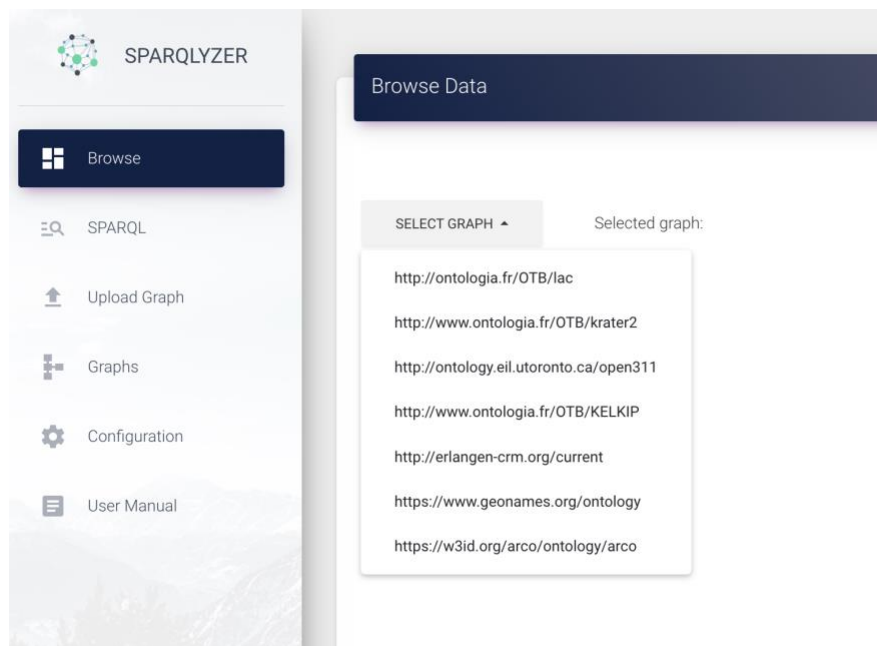


and the results will be available on the generated table. Every result is clickable and when clicked the selected uri opens in a new tab with its label, type and all properties with their domains [more details [here](#)]:

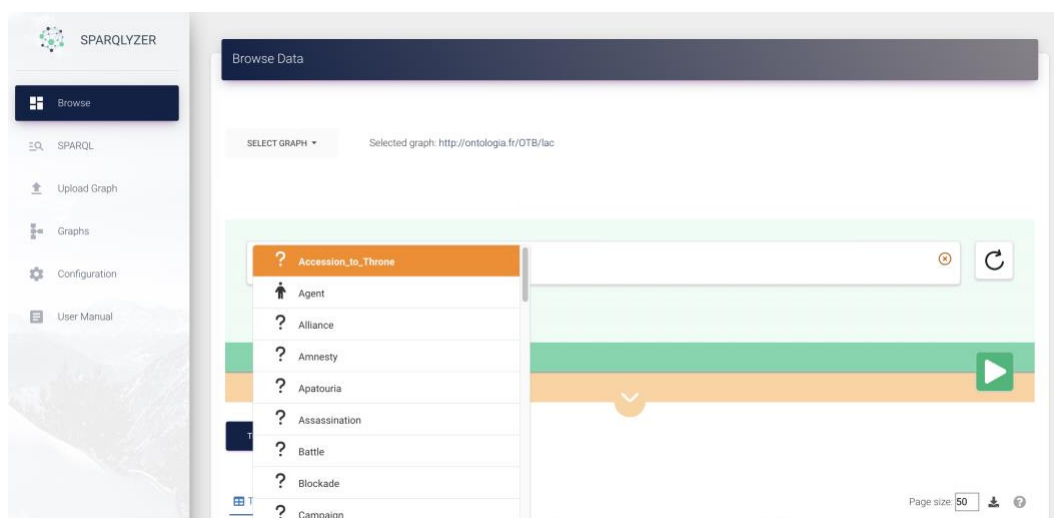
The image shows a web interface for exploring data. On the left is a sidebar with 'SPARQLYZER' and navigation options: Dashboard, SPARQL, Upload Graph, Graphs, and Configuration. The main content area is titled 'Explore Data' and shows details for the entity 'Ademantus'. The URI is 'http://ontologia.fr/OTB/lac#agi:Ademantus'. The type is 'NamedIndividual, Agent, Person'. There's a comment in Greek: 'Ademantus (Ancient Greek: Ἀδελμαντός), son of Leucolophides (Ancient Greek: Λευκολοφίδης), an Athenian, was one of the commanders with Alcibiades in the expedition against Andros in 407 BC.' There's also a comment in English: 'Ο Ἀδελμαντός (Ἀρχαία Ἑλληνικά: Ἀδελμαντός), γιος του Λευκολοφίδη (Ἀρχαία Ἑλληνικά: Λευκολοφίδης), ήταν Ἀθηναῖος και ένας από τους διοικητές μαζί με τον Ἀλκιβιάδη στην εκστρατεία κατά της Ἀνδρου το 407 π.Χ.' There are several 'seeAlso' links and a 'sonOf' link to 'http://ontologia.fr/OTB/lac#agi:Leucolophides'. At the bottom, there's a graph visualization showing a central node '1' connected to nodes '2', '3', '4', and '5' with labels 'sonOf', 'brother', 'brother', and 'brother' respectively.

## Browse Data

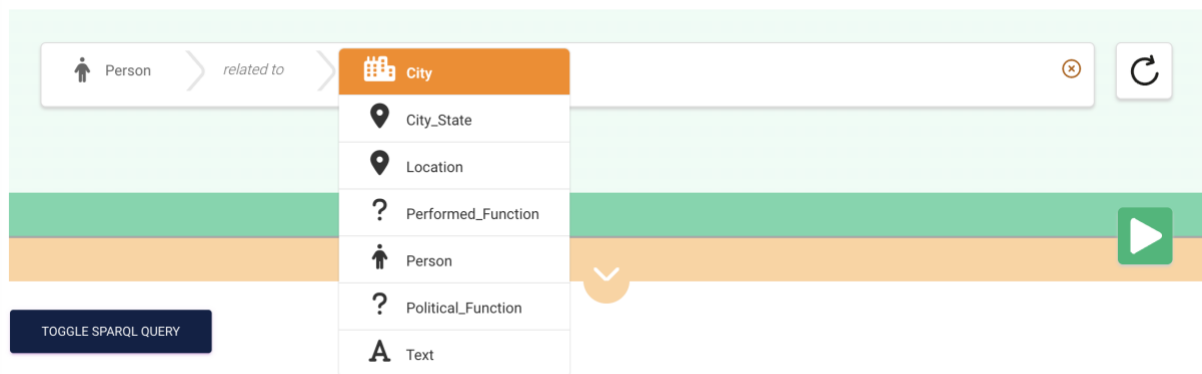
On the first screen the tool asks from the users to select a graph from a drop-down menu:



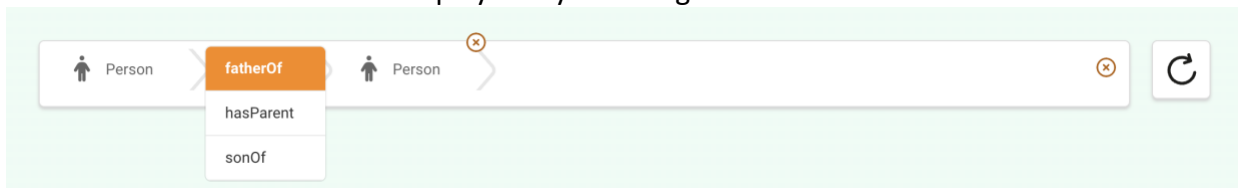
This drop-down list contains all the graphs that have already been uploaded to the triple store. Upon user's selection the tool dynamically creates all the classes and properties of the selected graph and sets the graphical query builder ([sparnatural](#)):



Initially, all the classes of the ontology are available. By selecting a class (e.g. Person):

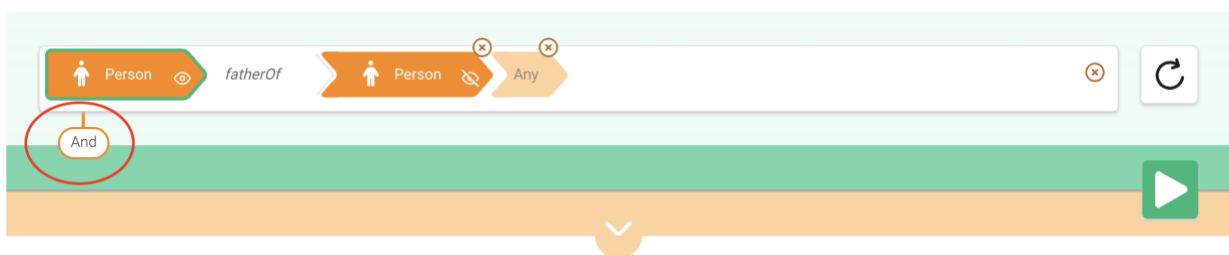


All the available domains are displayed. By selecting a domain:

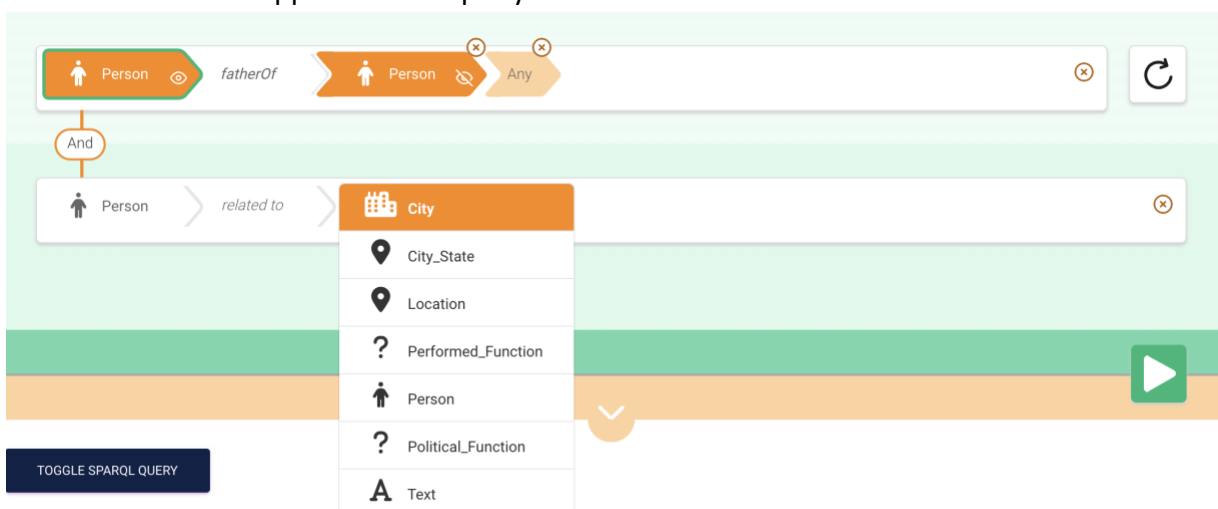


All available properties are displayed and users can choose what they want.

It is possible to add another class by clicking on this button:



And another class appears in the query builder:



It is also possible to add more domains of the query by clicking this button:

The screenshot shows a query builder interface with a light green background. At the top, there is a query bar containing: a Person icon, the text 'Person', a magnifying glass icon, the text 'fatherOf', another Person icon, a magnifying glass icon, and the text 'Any'. To the right of this bar is a refresh icon. Below this bar, an 'And' button is connected to a second query bar. This second bar contains: a Person icon, the text 'Person', a magnifying glass icon, the text 'bornIn', a magnifying glass icon, a City icon, a magnifying glass icon, and the text 'City'. Below the second bar, there is a search input field with the text 'Any(City) or Find:'. Below the search field, there are two numbered buttons: '1 Any(City) or Find:' and '2 Search City where...'. The '+' button in the second button is circled in red. At the bottom right of the interface is a green play button. A yellow arrow points downwards from the bottom of the screenshot.


And the query may be more complex:

The screenshot shows a query builder interface with a light green background. At the top, there is a query bar containing: a Person icon, the text 'Person', a magnifying glass icon, the text 'fatherOf', another Person icon, a magnifying glass icon, and the text 'Any'. To the right of this bar is a refresh icon. Below this bar, an 'And' button is connected to a second query bar. This second bar contains: a Person icon, the text 'Person', a magnifying glass icon, the text 'bornIn', a magnifying glass icon, a City icon, a magnifying glass icon, and the text 'City'. Below the second bar, there is a 'Where' button. Below the 'Where' button, there is a third query bar. This third bar contains: a City icon, a magnifying glass icon, the text 'related to', a magnifying glass icon, an Ethnic\_Group icon, a magnifying glass icon, and the text 'Ethnic\_Group'. Below the third bar, there is a dropdown menu with the text 'Text'. At the bottom right of the interface is a green play button. A yellow arrow points downwards from the bottom of the screenshot.

By clicking on the Toggle SPARQL QUERY the constructed query is visible to the user:


The screenshot shows a SPARQL query builder interface. At the top, there is a visual query graph with three main parts: 1. A triple pattern: Person (with a person icon) - fatherOf -> Person (with a person icon). 2. A triple pattern: Person (with a person icon) - bornIn -> City (with a city icon). 3. A triple pattern: City (with a city icon) - related to -> Ethnic\_Group (with a person icon). A 'Where' box is connected to the City and Ethnic\_Group patterns. Below the graph is a 'TOGGLE SPARQL QUERY' button. Below that is a text area containing the following SPARQL query:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 SELECT DISTINCT ?Person_1 ?Person_1_label WHERE {
4   ?Person_1 rdf:type <http://ontologia.fr/OTB/lac#Person>.
5   OPTIONAL {
6     ?Person_1 rdfs:label ?Person_1_label.
7     FILTER((LANG(?Person_1_label)) = "en")
8   }
9   ?Person_1 <http://ontologia.fr/OTB/lac#fatherOf> ?Person_1;
10  rdf:type <http://ontologia.fr/OTB/lac#Person>;
11  <http://ontologia.fr/OTB/lac#bornIn> ?City_3.
12  ?City_3 rdf:type <http://ontologia.fr/OTB/lac#City>.
13 }
14 LIMIT 1000
```

By clicking the  the query executes and the results appears on the table :

The screenshot shows the SPARQL query builder interface after execution. The visual query graph now shows: 1. A triple pattern: Person (with a person icon) - name -> Text (with a 'A' icon). 2. A triple pattern: Text (with a 'A' icon) -> Any (with an 'Any' icon). Below the graph is a 'TOGGLE SPARQL QUERY' button.

 Table  Response 87 results in 0.196 seconds

Page size:   

- Person\_1**
- 1 [Adeimantus](#)
  - 2 [Aemilius Paulus](#)
  - 3 [Aeropus](#)
  - 4 [Agesandridas](#)
  - 5 [Agesilaus I](#)
  - 6 [Agis II](#)
  - 7 [Alcibiades](#)

Every result is clickable and when clicked a new tab opens with the explore data details.

## Explore Data

When the Explore Data page opens, it visualizes the selected result by its URI, showing its label, type and all properties with their domains:

Explore Data  
uri : <http://ontologia.fr/OTB/lac#agt-Adeimantus>

(en) Adeimantus, (el) Αδείμαντος

Type: NamedIndividual, Agent, Person

comment: Adeimantus (Ancient Greek: Αδείμαντος), son of Leucolophides (Ancient Greek: Λευκολοφίδης), an Athenian, was one of the commanders with Alcibiades in the expedition against Andros in 407 BC.

comment: Ο Αδείμαντος (Αρχαία Ελληνικά: Αδείμαντος), γιος του Λευκολοφίδη (Αρχαία Ελληνικά: Λευκολοφίδης), ήταν Αθηναίος και ένας από τους διοικητές μαζί με τον Αλκιβιάδη στην εκστρατεία κατά της Άνδρου το 407 π.Χ..

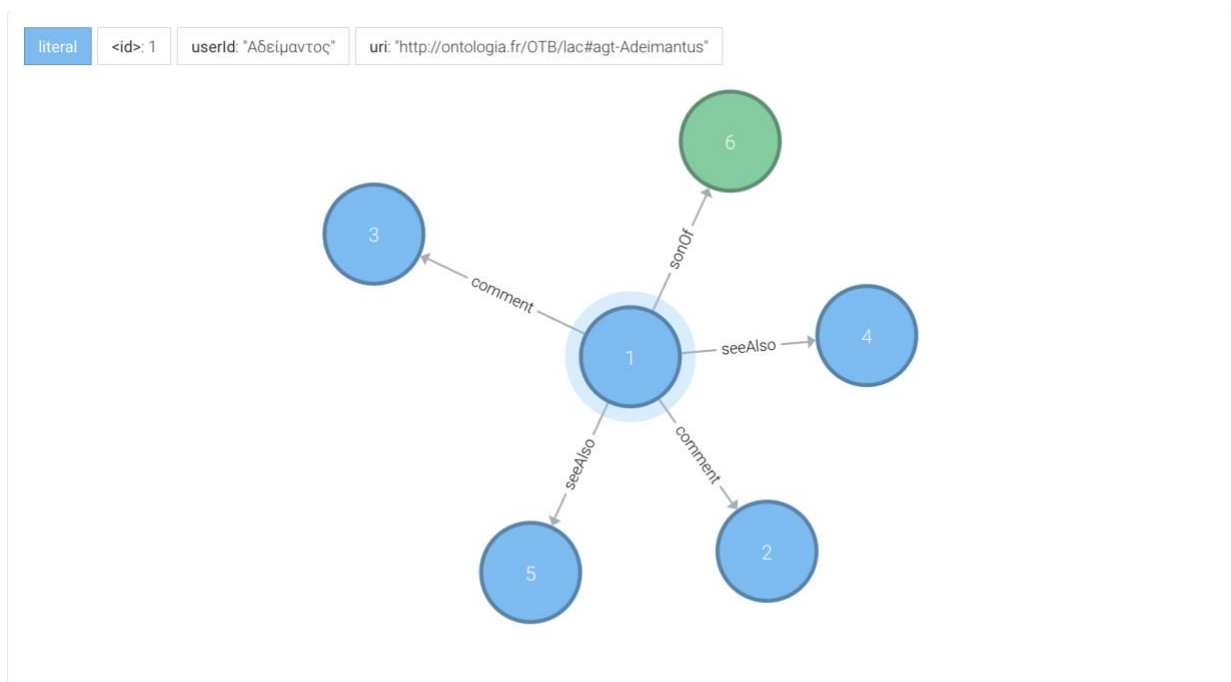
seeAlso: 2. The son of Leucolophides, an Athenian, was one of the commanders with Alcibiades in the expedition against Andros, B. C. 407. (Xen. Hell. 1.4.21.) He was again appointed one of the Athenian generals after the battle of Arginusae, B. C. 406, and continued in office till the battle of Aegospotami, B. C. 405, where he was one of the commanders, and was taken prisoner. He was the only one of the Athenian prisoners who was not put to death, because he had opposed the decrec for cutting off the right hands of the Lacedaemonians who might be taken in the battle. He was accused by many of treachery in this battle, and was afterwards impeached by Conon. (Xen. Hell. 1.7.1, 2.1.30-32; Paus. 4.17.2, x. § 5; Dem. de fals. leg. p. 401.; Lys. c. Alc. pp. 143, 21.) Aristophanes speaks of Adeimantus in the "Frogs" (1513), which was acted in the year of the battle, as one whose death was wished for; and he also calls him, apparently out of jest, the son of Leucolophus, that is, "White Crest." In the " Protagoras" of Plato, Adeimantus is also spoken of as present on that occasion (p. 315c.). William Smith. A Dictionary of Greek and Roman biography and mythology. London. John Murray: printed by Spottiswoode and Co., New-Street Square and Parliament Street. In the article on Soranus, we find: "at this present time (1848)" and this date seems to reflect the dates of works cited. 1873 - probably the printing date. <https://www.perseus.tufts.edu/hopper/text?doc=Perseus:text:1999.04.0104:entry=adeimantus-bio-2&highlightauth=-407#match1>

seeAlso: [https://en.wikipedia.org/wiki/Adeimantus\\_\(son\\_of\\_Leucolophides\)](https://en.wikipedia.org/wiki/Adeimantus_(son_of_Leucolophides))

sonOf: <http://ontologia.fr/OTB/lac#agt-Leucolophides>

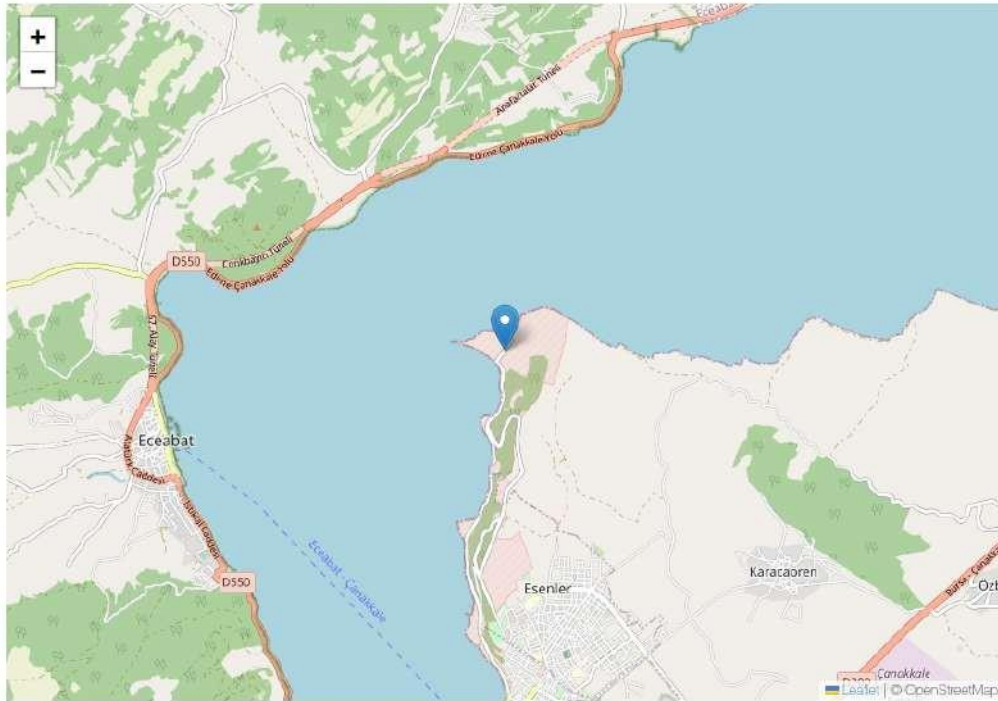
If domain values are links, they are clickable and users can extend the exploration to new tabs.

There is another visualization using the [neo4jd3](#) js library.



(Future work will make this graph dynamically expandable to allow users to explore data graphically).

Where domains contain geographical coordinates, the Explore Data page will display these coordinates as a pin on a [leaflet](#) map:



And if the property has a domain with a working image link, the tool is able to display it directly on the page:

Explore Data  
uri : <http://www.ontologia.fr/OTB/krater2#Beazley-16377>

(en) Beazley-16377

Type: NamedIndividual,  
Vessel\_for\_mixing\_wine\_with\_water\_with\_handles\_with\_foot\_with\_open\_mouth\_without\_neck\_with\_upward\_curling\_handles\_upward\_curling\_handles\_placed\_low\_on\_the\_body, Object

seeAlso: <http://ark.dasch.swiss/ark:/72163/080e-75aaae260b2e1-c>

seeAlso: <http://www.beazley.ox.ac.uk/record/3FC608C0-97AA-470C-A2FB-4815FB62EB7B>

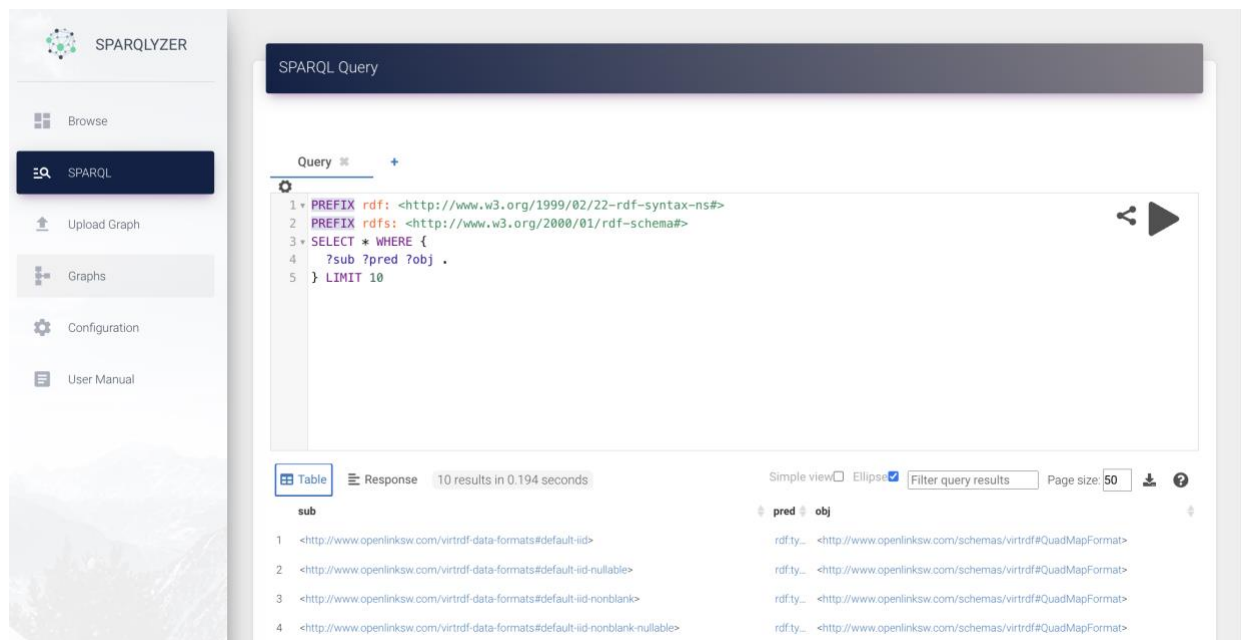
prefLabel: Beazley-16377

<http://xmlns.com/foaf/0.1/depiction: http://ontologia.fr/Images/GreekVase/Beazley-16377.png>



# SPARQL

On SPARQL query tab the tool provides a typical SPARQL endpoint using the [yasgui](#) library



The screenshot shows the SPARQLYZER web interface. On the left is a sidebar with navigation options: Browse, SPARQL (selected), Upload Graph, Graphs, Configuration, and User Manual. The main area is titled 'SPARQL Query' and contains a query editor with the following code:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 SELECT * WHERE {
4   ?sub ?pred ?obj .
5 } LIMIT 10
```

Below the query editor, the results are displayed in a table view. The table has two columns: 'sub' and 'obj'. The first column contains URIs, and the second column contains RDF types. The results are:

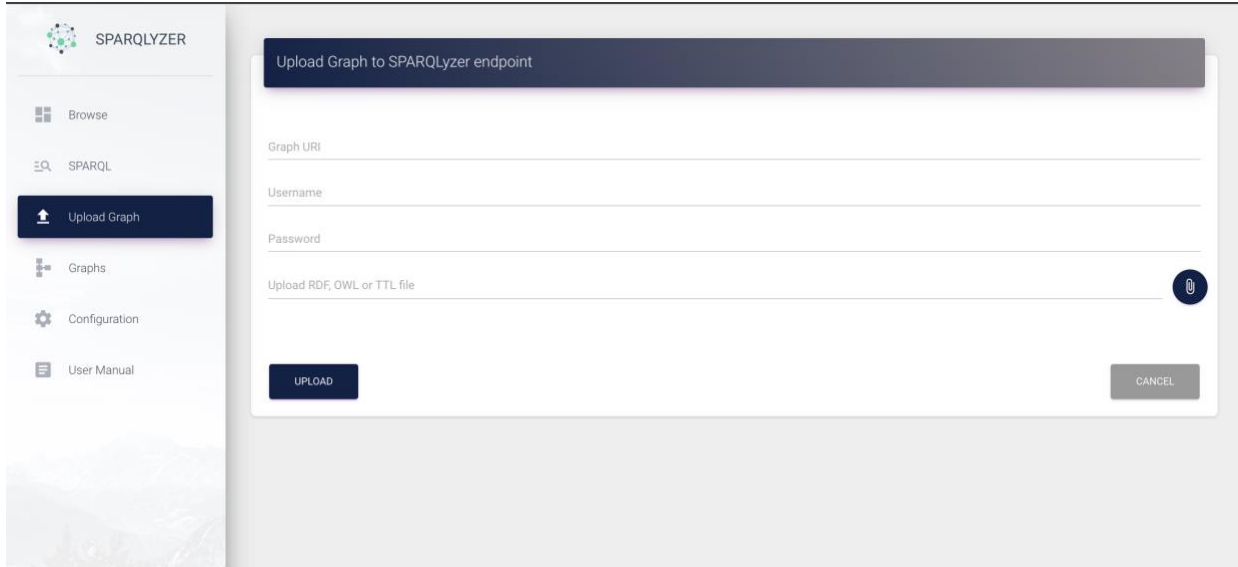
| sub   | obj   |
|---|---|
| <a href="http://www.openlinksw.com/virtrdf-data-formats#default-iiid">http://www.openlinksw.com/virtrdf-data-formats#default-iiid</a>                                     | <a href="http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat">http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat</a> |
| <a href="http://www.openlinksw.com/virtrdf-data-formats#default-iiid-nullable">http://www.openlinksw.com/virtrdf-data-formats#default-iiid-nullable</a>                   | <a href="http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat">http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat</a> |
| <a href="http://www.openlinksw.com/virtrdf-data-formats#default-iiid-nonblank">http://www.openlinksw.com/virtrdf-data-formats#default-iiid-nonblank</a>                   | <a href="http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat">http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat</a> |
| <a href="http://www.openlinksw.com/virtrdf-data-formats#default-iiid-nonblank-nullable">http://www.openlinksw.com/virtrdf-data-formats#default-iiid-nonblank-nullable</a> | <a href="http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat">http://www.openlinksw.com/schemas/virtrdf#QuadMapFormat</a> |

The interface also shows a status bar indicating '10 results in 0.194 seconds' and a 'Filter query results' input field.

Users can create their own SPARQL queries and execute them in a similar way as described above. As described, all the results are clickable and can be visualized with the [explore data](#) features.


## Upload Graph

The tool allows graph upload (.rdf, .owl, .ttl files allowed) from local computers on the dedicated repository.



The screenshot displays the SPARQLYZER web interface. On the left is a sidebar with a menu containing 'Browse', 'SPARQL', 'Upload Graph' (which is highlighted in dark blue), 'Graphs', 'Configuration', and 'User Manual'. The main content area is titled 'Upload Graph to SPARQLyzer endpoint'. It features four input fields: 'Graph URI', 'Username', 'Password', and 'Upload RDF, OWL or TTL file' (which includes a file upload icon). At the bottom of the form are two buttons: 'UPLOAD' and 'CANCEL'.

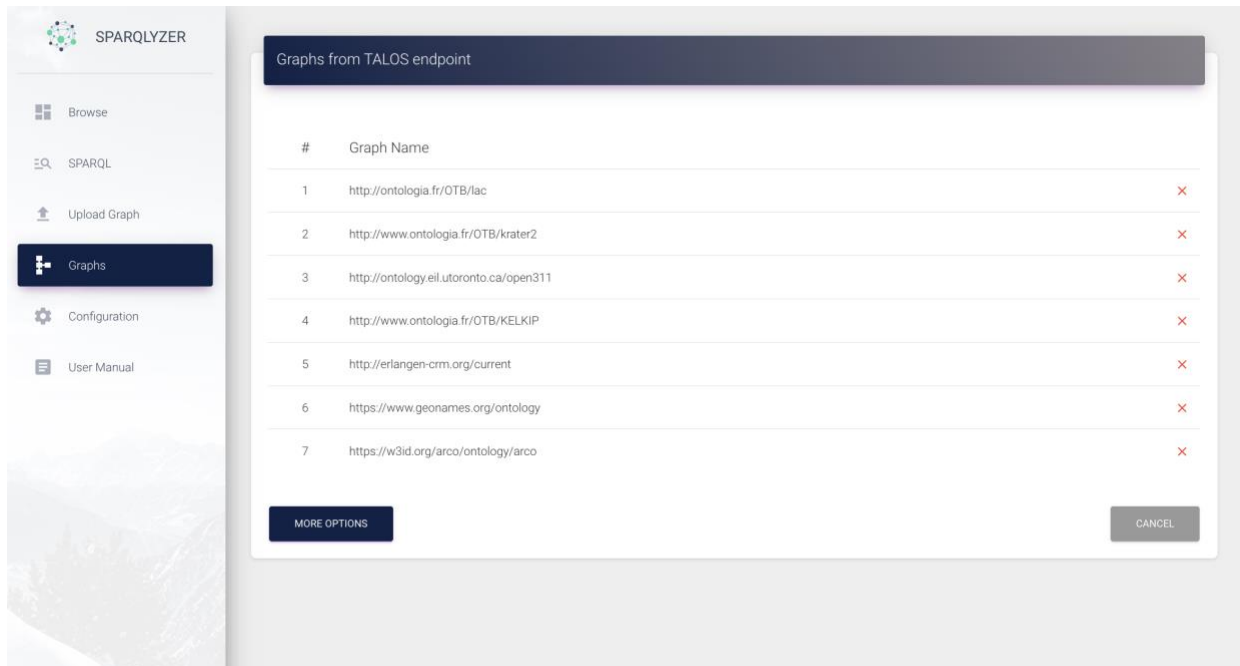
Users must fill all the above inputs properly to upload their graphs:

- Graph URI: the URI of the uploaded ontology **as it appears in the data** (eg. For the LACRIMALit ontology the graph URI should be `http://ontologia.fr/OTB/lac`).
- Triplestore username and password (credentials will be sent to eligible users)
- Upload the RDF data from your computer by clicking this  button.
- Click on the Upload button.

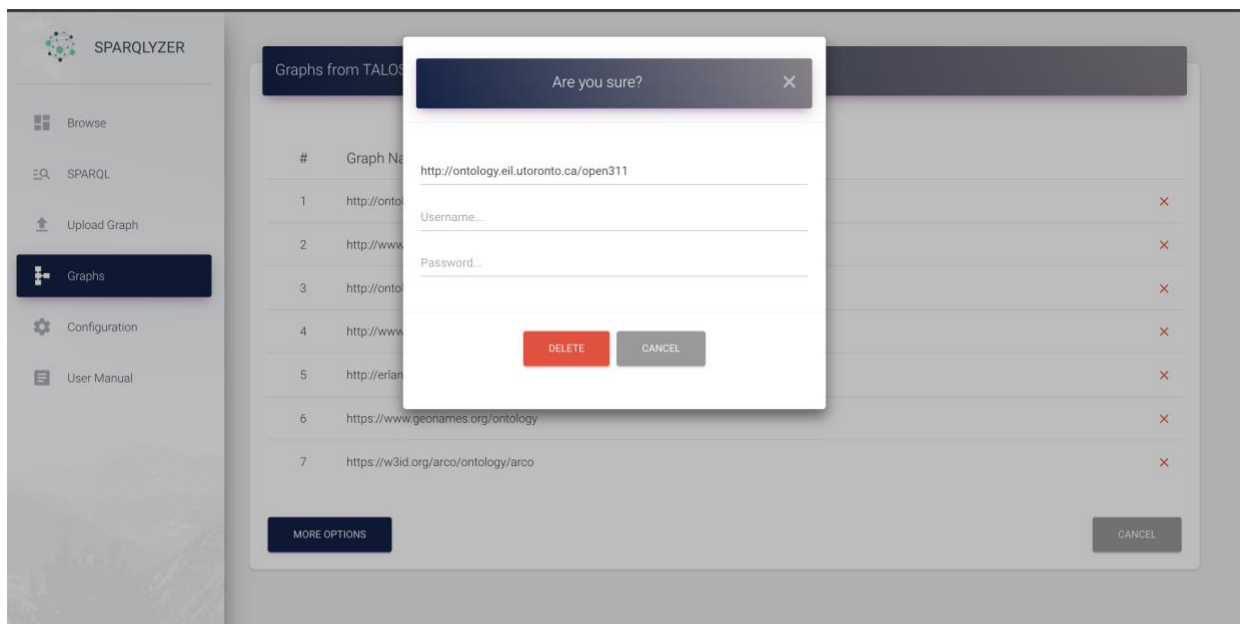
If all the above fields are filled in correctly, the graph will be ready to start querying.

# Graphs

In this tab all the uploaded graphs are displayed as below:

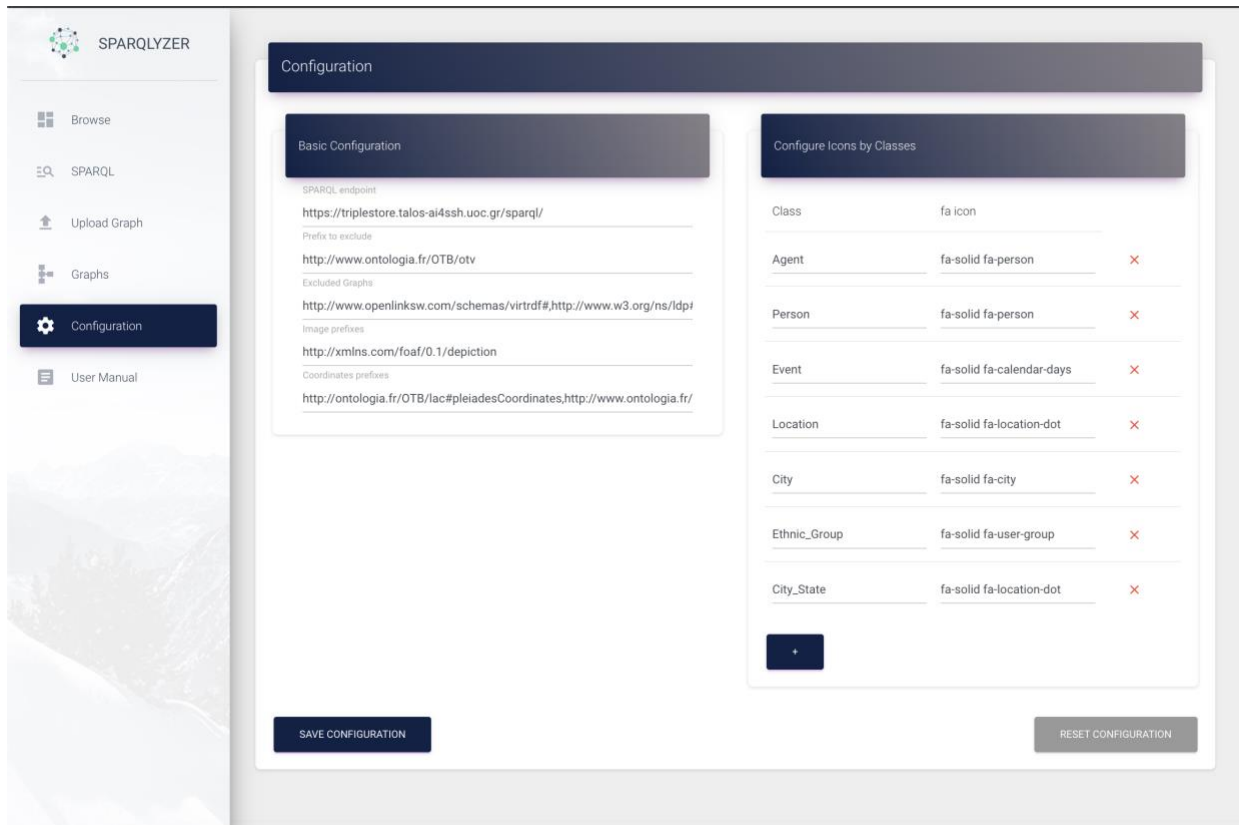


Clicking this **X** button users are able to delete the selected graphs. When this button is clicked, a modal shows up and asks confirmation of the action requiring the triple store credentials:



# Configuration

By default, the tool has all the necessary configurations predefined to work properly. It allows users to create their own configuration by editing the inputs below:



## Basic Configuration:

- SPARQL endpoint : this field must contain the default URL of the endpoint where all actions (uploading data, running queries) will be performed.
- Prefix to exclude: If an ontology has more uris which contain classes except the initial uri, these uris must be filled in this field in order to create all the classes properly.
- Excluded Graphs: This field must contain any default triple store graphs that the user does not want to be visible on the Graphs tab (eg. <http://www.openlinksw.com/schemas/virttrdf#>, <http://www.w3.org/ns/ldp#> and other default graphs of each triple store may includes).
- Image Prefixes: This field must contain any image prefixes that may have image urls as a range in order for the image url to be visible as an image on the Explore Data page.
- Coordinates prefixes: This field must contain any urls that may have coordinates or other geographic data as a range in order for the image url to be visible on maps in the Explore Data page.

## Configure Icons by Classes:


This tab allows the user to define their preferences in order to customize the icons of the classes on the initial sparnatural screen:



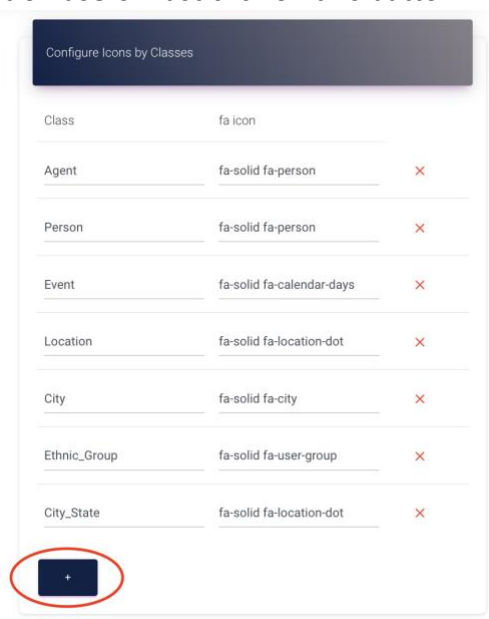
If a class has been configured with the desired icon, this icon will be displayed on the above screen.

For instance:

| Class | fa icon            |   |
|-------|--------------------|---|
| Agent | fa-solid fa-person | ✖ |

Class Agent has the icon of fa-solid fa-person and this class will be depicted with this  icon.

In order to add a new depiction users must click on this button :



And a new row will be created:

|   |  |                                |
|---|--|--------------------------------|
| <input type="text" value="Ethnic_Group"/> | <input type="text" value="fa-solid fa-user-group"/>          | <input type="text" value="✘"/> |
| <input type="text" value="City_State"/>   | <input type="text" value="fa-solid fa-location-dot"/>        | <input type="text" value="✘"/> |
| <input type="text" value="Enter Class"/>  | <input type="text" value="Enter icon value eg. fa-solid f"/> | <input type="text" value="✘"/> |

There user must enter the class and the desired icon values using the [fontawesome](#) icon library.

## Contact

Kostas Petrakis <[petrakis1@gmail.com](mailto:petrakis1@gmail.com)>

