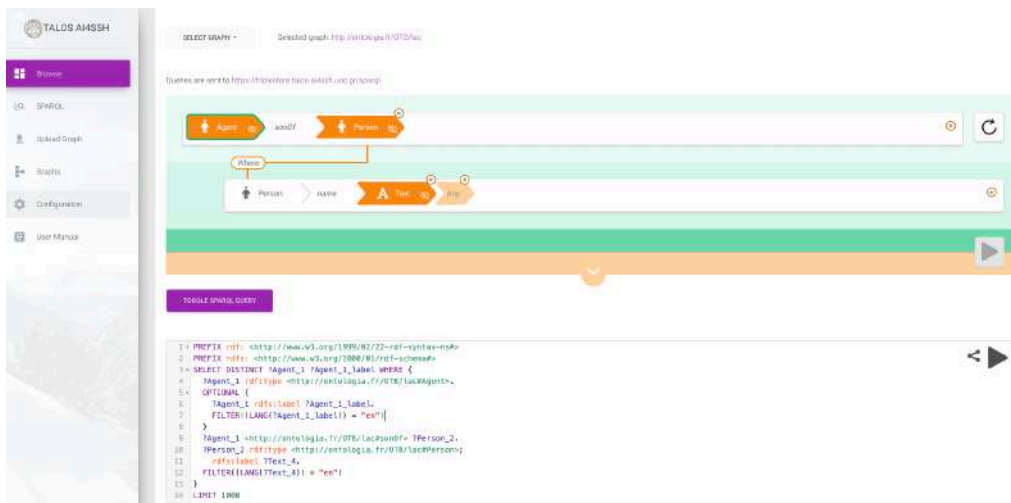


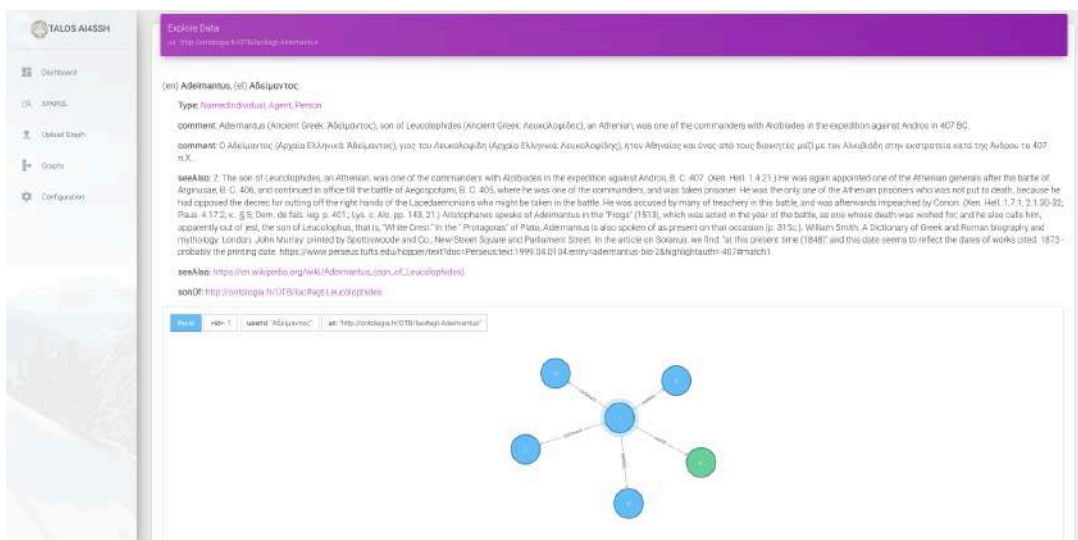
User Manual: TALOS Dashboard

Description



TALOS Dashboard 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 the TALOS dedicated [triplestore](#)
- Query the TALOS SPARQL endpoint
- Delete graphs from the triplestore




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](#).

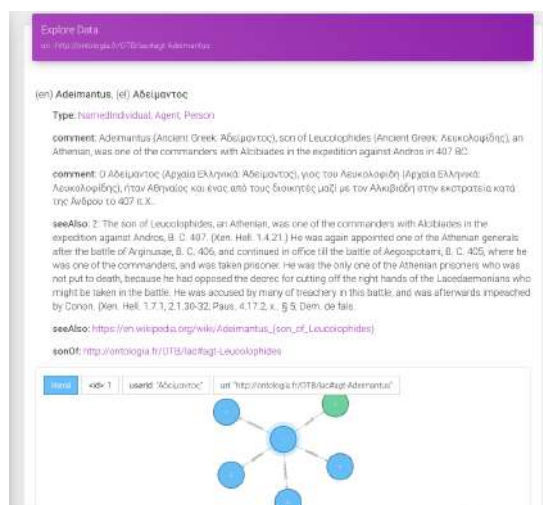
Quick Start

The following steps must be taken for a 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](#)]:

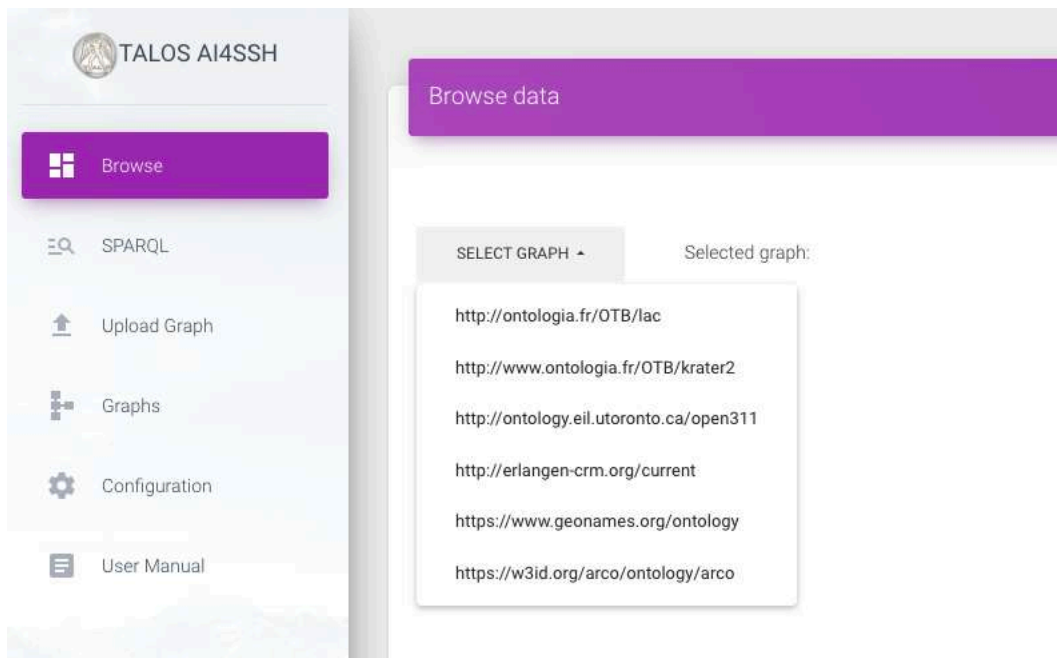


and the results will be available on the generated table. Every result is clickable and when clicked the selected uri opens in a new tab with its label, type and all properties with their domains [more details [here](#)]:

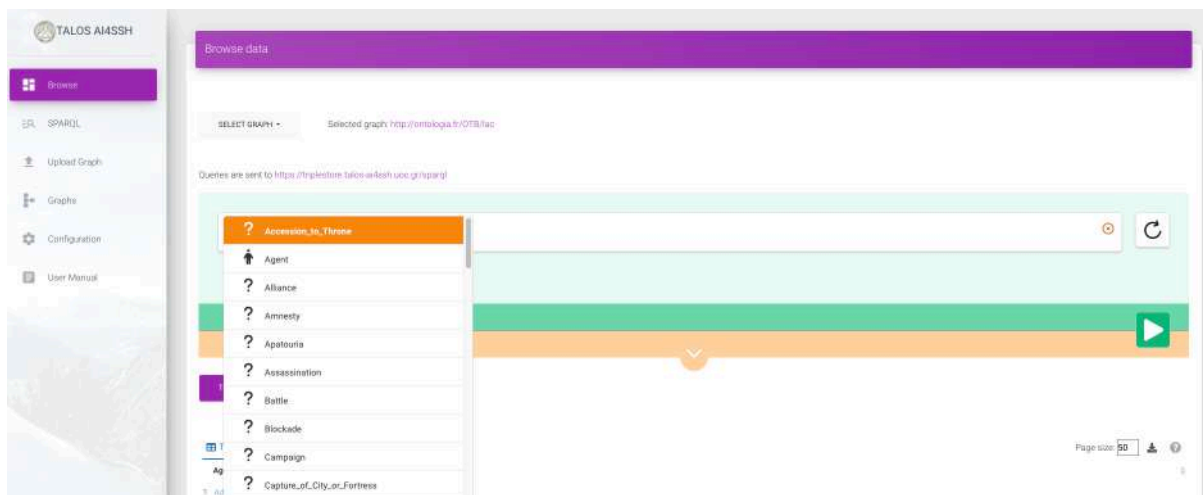


Browse Data

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:

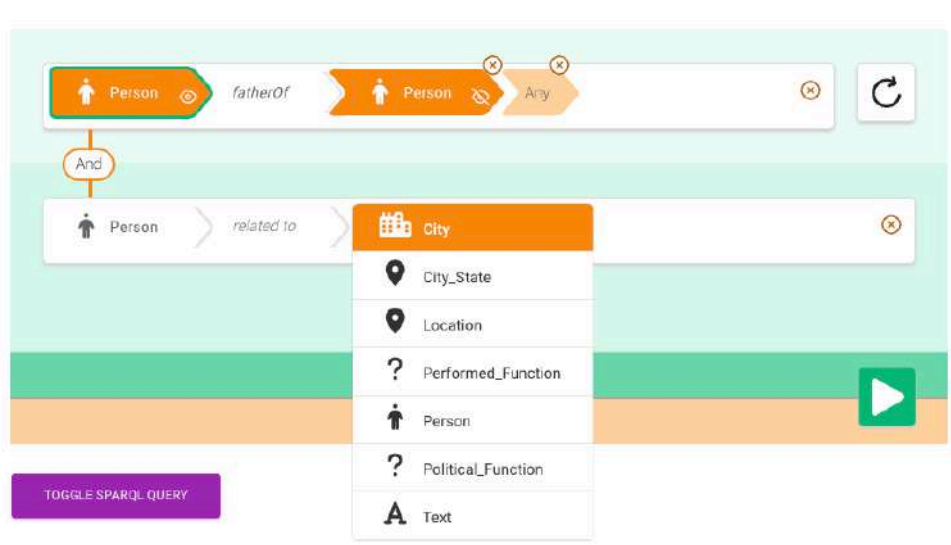


All available properties are displayed and users can choose what they want.

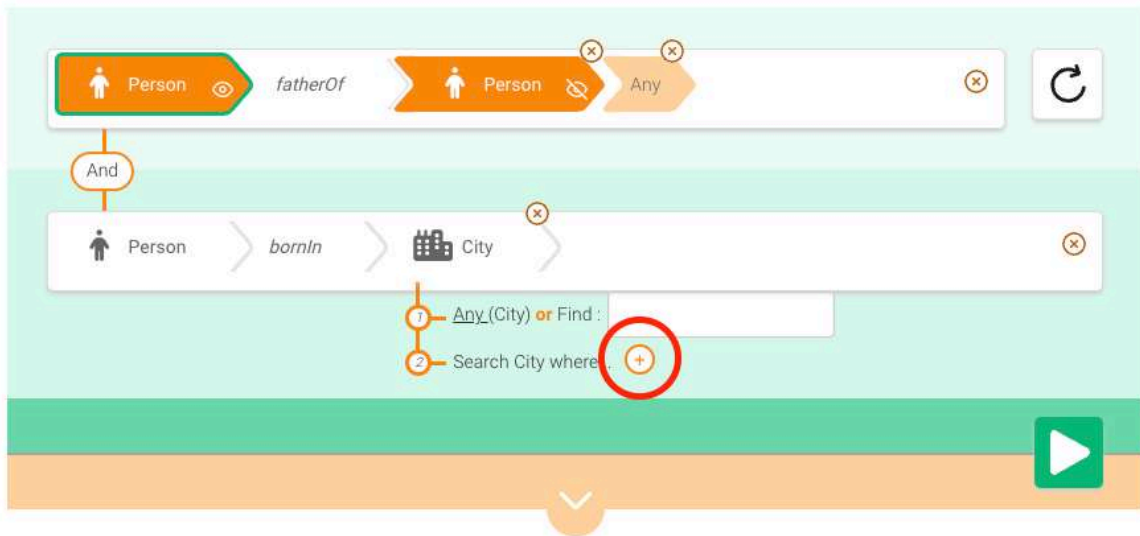
It is possible to add another class by clicking on this button:



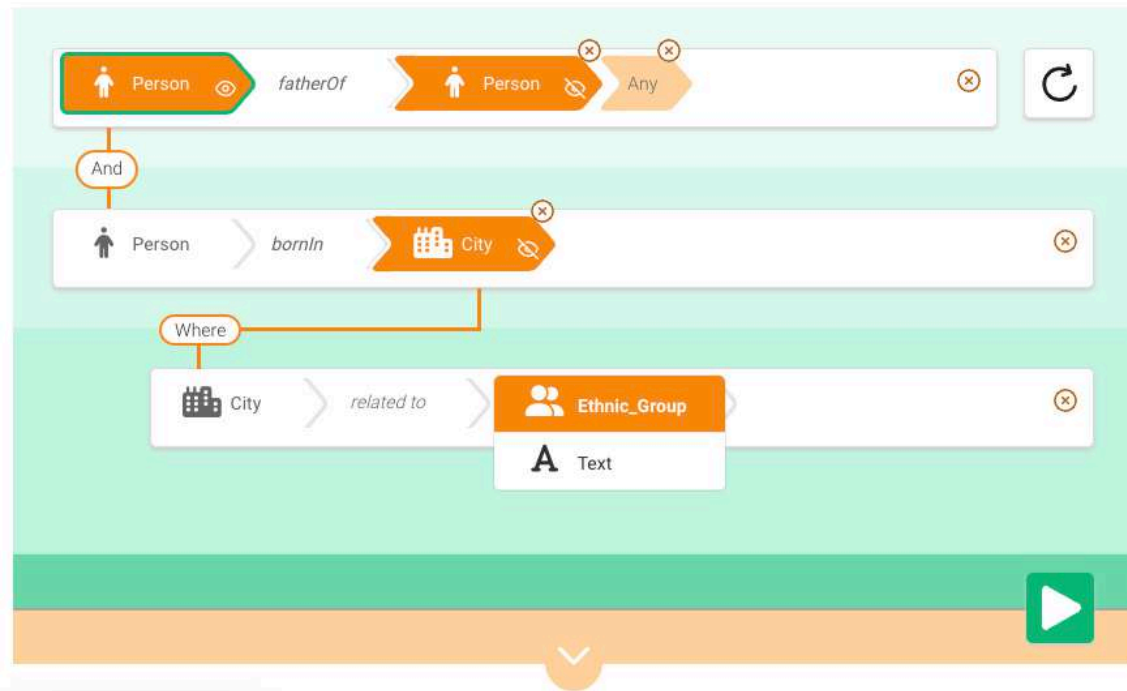
And another class appears in the query builder:



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




And the query may be more complex:



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

The screenshot shows a visual query builder interface. At the top, there is a query graph with three main components: 1. A 'Person' node connected to a 'fatherOf' property, which is connected to another 'Person' node. 2. A 'Person' node connected to a 'bornIn' property, which is connected to a 'City' node. 3. A 'City' node connected to a 'name' property, which is connected to a 'Text' node. The graph is connected by 'And' and 'Where' operators. Below the graph is a 'TOGGLE SPARQL QUERY' button. The bottom part of the image shows the resulting SPARQL query code in a text area.

```
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://ontology.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://ontology.fr/OTB/lac#fatherOf> ?Person_2.
10  ?Person_2 rdf:type <http://ontology.fr/OTB/lac#Person>.
11  ?Person_1 <http://ontology.fr/OTB/lac#bornIn> ?City_4.
12  ?City_4 rdf:type <http://ontology.fr/OTB/lac#City>;
13    rdfs:label ?Text_6.
14  FILTER((LANG(?Text_6)) = "en")
15 }
16 LIMIT 1000
```

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

The screenshot shows the same SPARQL query builder interface, but now the query has been executed. The 'TOGGLE SPARQL QUERY' button is still visible. Below it, a table displays the results. The table has two columns: 'Agent_1' and 'Response'. The first column contains the names of the agents, and the second column contains their corresponding labels. The results are listed in a numbered order from 1 to 10.

Agent_1	Response
1	Adelmantax
2	Aernillus Paulus
3	Aeropus
4	Agessandinos
5	Agelkazo I
6	Agia II
7	Aicbiades
8	Aicidas
9	Alexander, son of Aeropus
10	Alexander III king of Macedon

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://ontology.fr/OTB/lac#agt-Adeimantus>

(en) Adeimantus, (el) Αδείμαντος

Type: [NamedIndividual](#), [Agent](#), [Person](#)

comment: Adeimantus (Ancient Greek: Ἀδείμαντος), 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: Ἀδείμαντος), 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..

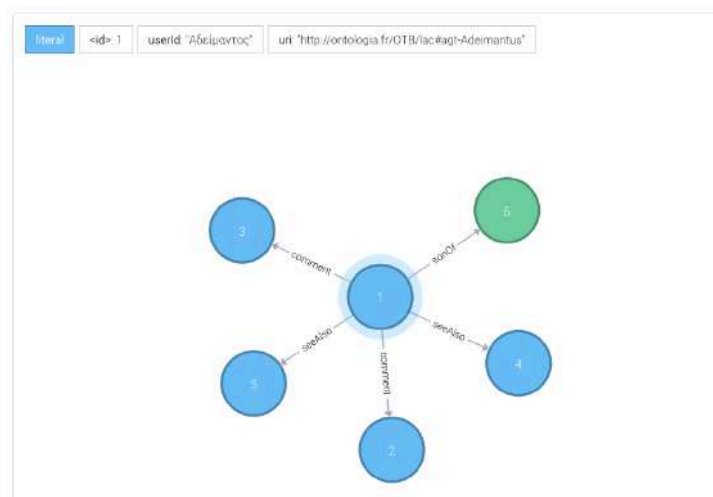
seeAlso: 2. The son of Leucolophides, an Athenian, was one of the commanders with Alcibiades in the expedition against Andros, B. C. 407. (Xen. Hell. 1.4.21.) He was again appointed one of the Athenian generals after the battle of Arginusae, B. C. 406, and continued in office till the battle of Aegospotami, 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 decree 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://ontology.fr/OTB/lac#agt-Leucolophides>

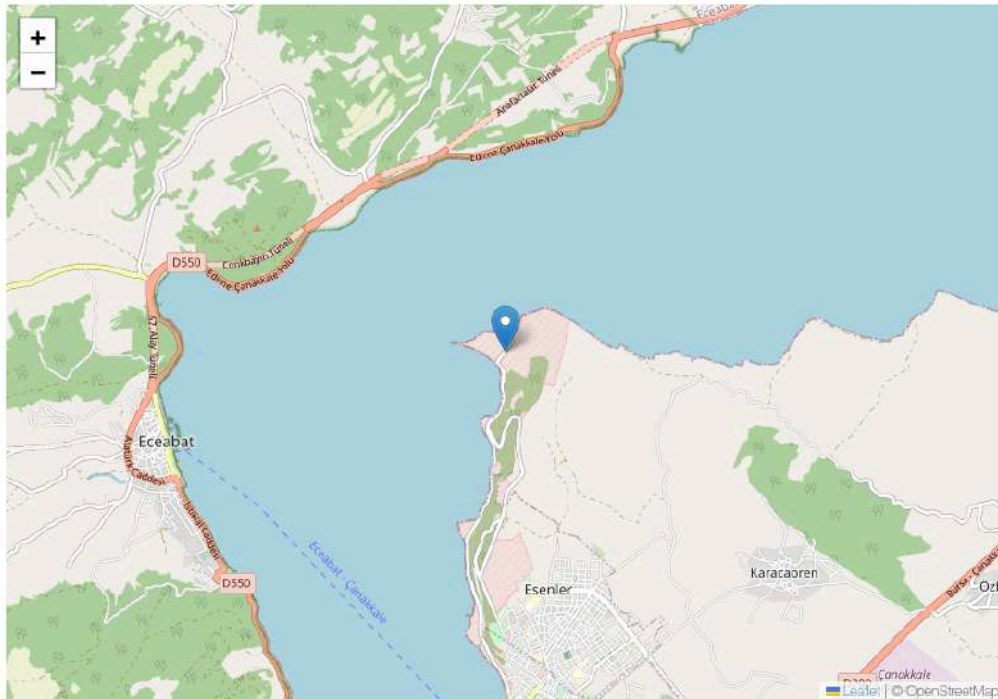
If domain values are links, they are clickable and users can extend the exploration to new tabs.

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/3FC608CD-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 displays the SPARQL Query interface in TALOS AI4SSH. The interface is divided into a sidebar on the left and a main content area on the right. The sidebar contains navigation options: Browse, SPARQL (selected), Upload Graph, Graphs, Configuration, and User Manual. The main content area features a purple header for 'SPARQL Query' and a query editor with 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 ?object_1 ?object_1_label WHERE {
4   ?object_1 rdfs:type <http://www.ontologia.fr/OTB/otv#Object>.
5   OPTIONAL {
6     ?object_1 rdfs:label ?object_1_label.
7     FILTER((LANG(?object_1_label)) = "en")
8   }
9   ?object_1 rdfs:label ?Text_2.
10  FILTER((LANG(?Text_2)) = "en")
11 }
12 LIMIT 1000
```

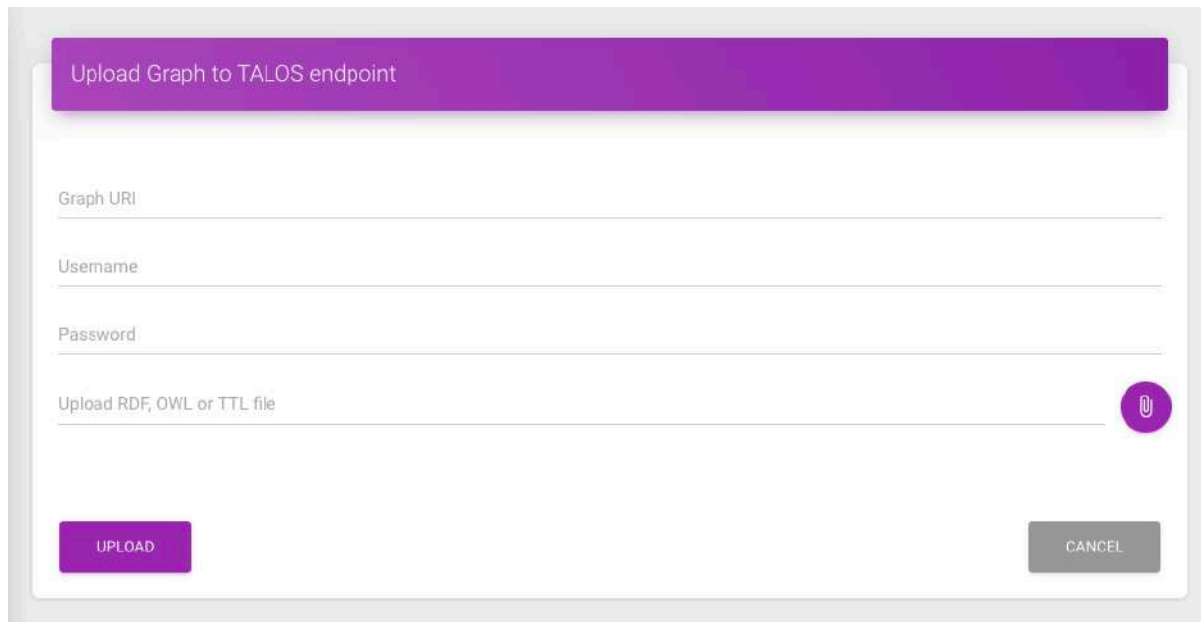
Below the query editor, the results are displayed in a table format. The table has two columns: 'Object_1' and 'Object_1_label'. The results are as follows:

Object_1	Object_1_label
<http://www.ontologia.fr/OTB/krater2#Bezley-305051>	'Bezley-305051' ^{en}
<http://www.ontologia.fr/OTB/krater2#Bezley-10934>	'Bezley-10934' ^{en}
<http://www.ontologia.fr/OTB/krater2#Bezley-10936>	'Bezley-10936' ^{en}
<http://www.ontologia.fr/OTB/krater2#Bezley-11089>	'Bezley-11089' ^{en}
<http://www.ontologia.fr/OTB/krater2#Bezley-15412>	'Bezley-15412' ^{en}

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 TALOS dedicated [repository](#).



The screenshot shows a web form titled "Upload Graph to TALOS endpoint". The form contains four input fields: "Graph URI", "Username", "Password", and "Upload RDF, OWL or TTL file". The "Upload RDF, OWL or TTL file" field has a purple circular button with a white paperclip icon. At the bottom of the form, there are two buttons: a purple "UPLOAD" button and a grey "CANCEL" button.

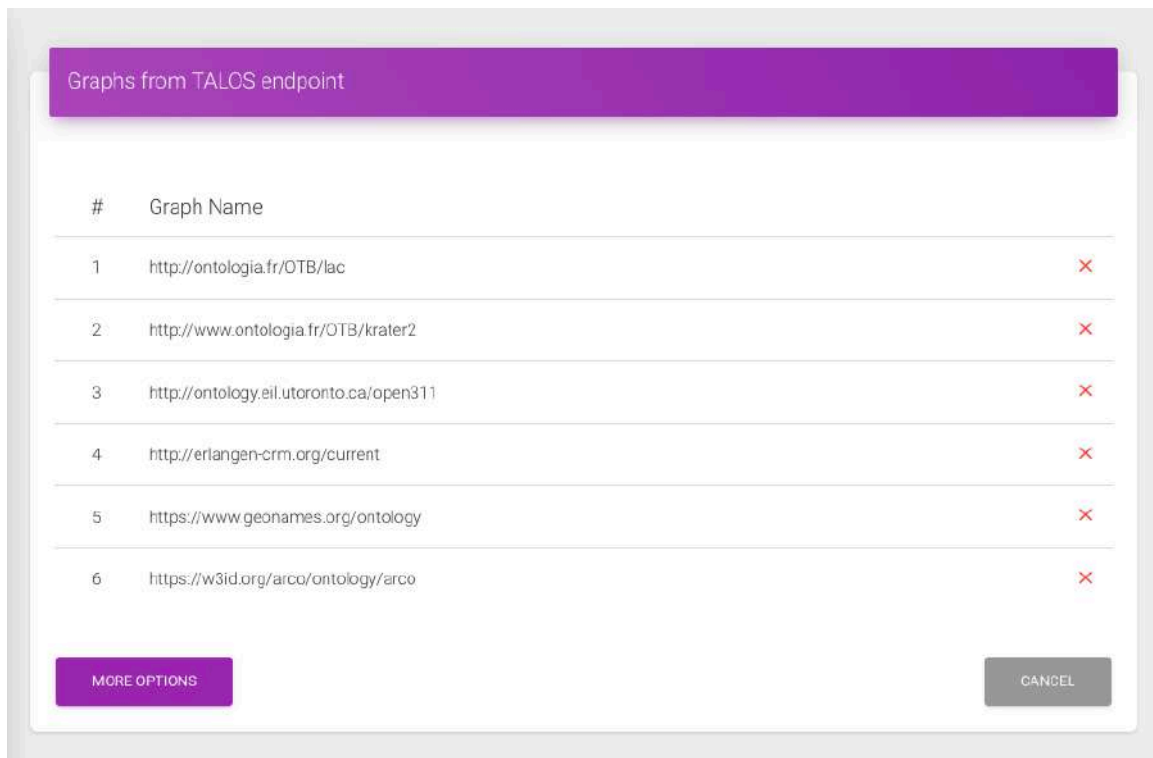
Users must fill all the above inputs properly to upload their graphs:

- 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.

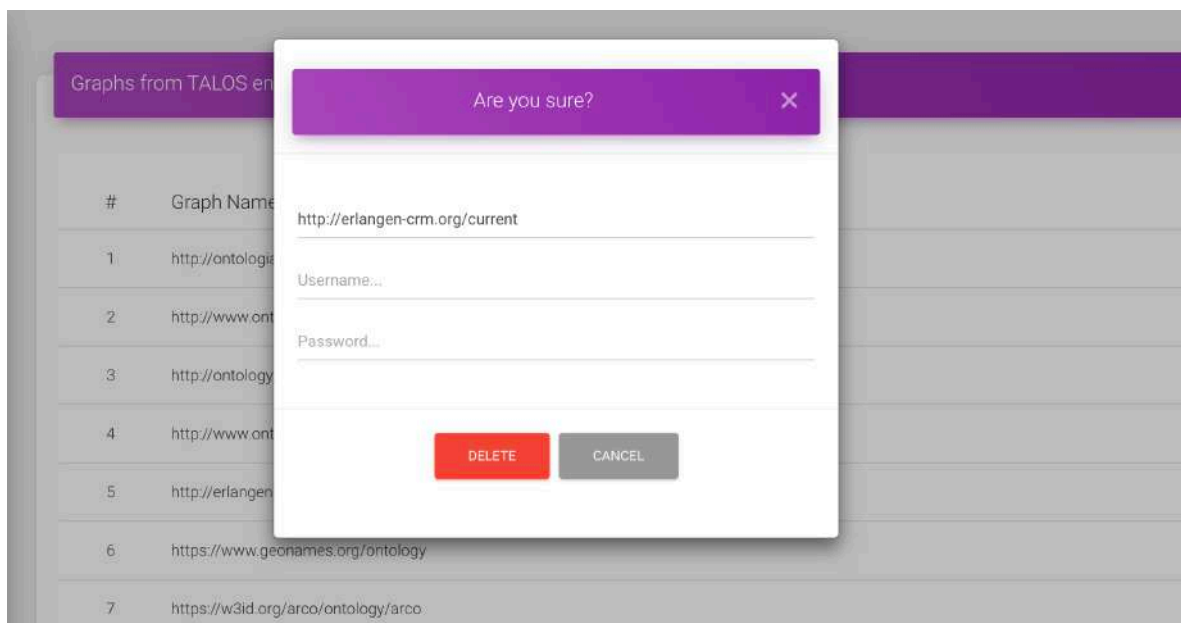
If all the above fields are filled in correctly, the graph will be ready to start querying.

Graphs

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



Clicking this ✘ 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:

The screenshot displays a configuration interface with two main sections:

- Basic Configuration:** Contains input fields for SPARQL endpoint, Prefix to exclude, Excluded Graphs, Image prefixes, and Coordinates prefixes.
- Configure Icons by Classes:** A table with columns for Class, Icon, and a delete button (X).

Class	Icon	
Agent	fa-solid fa-person	X
Person	fa-solid fa-person	X
Event	fa-solid fa-calendar-d	X
Location	fa-solid fa-location-dc	X
City	fa-solid fa-city	X
Ethnic_Group	fa-solid fa-user-group	X
City_State	fa-solid fa-location-dc	X

At the bottom, there are two buttons: "SAVE CONFIGURATION" and "RESET CONFIGURATION".

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:



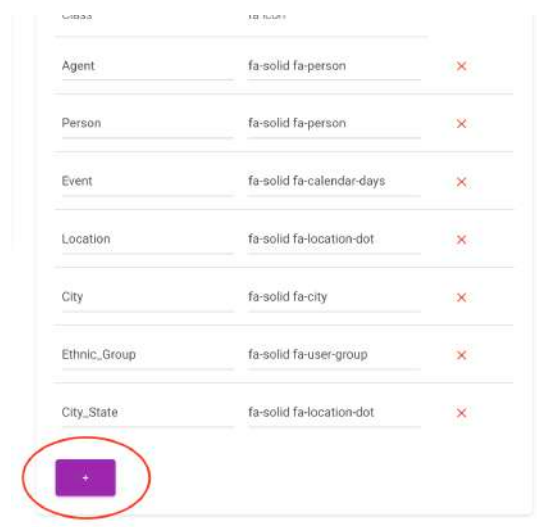
If a class has been configured with the desired icon, this icon will be displayed on the above screen.

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:

Ethnic_Group	fa-solid fa-user-group	✘
City_State	fa-solid fa-location-dot	✘
Enter Class	Enter icon value eg. fa-solid fa-	✘



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

Contact

Kostas Petrakis < cpetrak@uoc.gr >