

EcoPortal Documentation

	Name:	Title:
Authors:	Lucia Vaira	LifeWatch ERIC Service Centre ICT Coordinator
	Ilaria Rosati	National Research Council of Italy, Research Institute on Terrestrial Ecosystems, Technologist
	Enrica Nestola	National Research Council of Italy, Research Institute on Terrestrial Ecosystems, Technologist
	Cristina Di Muri	National Research Council of Italy, Research Institute on Terrestrial Ecosystems, Technologist
	Alexandra Nicoleta Muresan	National Research Council of Italy, Research Institute on Terrestrial Ecosystems, Technologist
	Davide Raho	National Research Council of Italy, Research Institute on Terrestrial Ecosystems, Technologist
	Majid Fiaz	LifeWatch ERIC Web Developer
	Parham Ramezani	LifeWatch ERIC ICT Ontology Junior Developer
	Martina Pulieri	University of Salento, Research Fellow
Reviewer:	Cosimo Vallo	Training Officer
Approver:	Alberto Basset	Interim Director of Service Centre

Table of Contents

1.	About EcoPortal.....	4
1.1	What is EcoPortal and who developed it?.....	4
1.2	Why EcoPortal and how it works?.....	4
2.	EcoPortal overview	4
2.1	Homepage.....	4
2.1.1	Navigation bar	5
2.1.2	Account settings.....	5
2.1.3	O'FAIRe tool	7
2.1.4	Footer.....	7
2.2	EcoPortal tools.....	8
2.2.1	Browse.....	8
2.2.2	Search.....	10
2.2.3	Annotator.....	11
2.2.4	Mappings.....	12
2.2.5	Recommender.....	15
2.2.6	VocBench.....	16
2.3	Semantic artefact detail page.....	19
2.3.1	Summary.....	21
2.3.2	Menu.....	26
3.	EcoPortal Metadata Schema.....	4
4.	Semantic Artefact lifecycle	4
4.1	Create and edit semantic artefacts.....	72
4.2	Export a semantic artefact from VocBench.....	4
4.3	Submit new semantic artefacts	4
4.4	Update semantic artefacts	4
4.5	Give feedback on semantic artefacts	4

4.6	SKOS Support	4
5.	Developer guide	4
5.1	API key.....	93
5.2	REST API Documentation.....	95
5.2.1	Common Parameters	95
5.2.2	Resource Endpoints.....	97
5.2.3	Service endpoints.....	128
5.3	Using EcoPortal technology in your project.....	134
5.4	EcoPortal widgets: how to use them.....	135
6.	Document Revision History	4

1. About EcoPortal

1.1 What is EcoPortal and who developed it?

EcoPortal is a repository of semantic artefacts in the ecological domain. EcoPortal was developed by [LifeWatch ERIC](#), in collaboration with its Italian node [LifeWatch Italy](#), coordinated by the [National Research Council](#), through the reuse of the OntoPortal technology to support the activities of different kinds of stakeholders (researchers, vocabulary managers, data managers, ontologists, etc.). EcoPortal assists the community in the creation, management, mapping and alignment of FAIR semantic artefacts, which make the meaning of research data clear and explicit, ultimately increasing their impact.

1.2 Why EcoPortal and how it works?

In the last decade, ecological research groups and research infrastructures have contributed to the production of numerous vocabularies and ontologies to increase data discovery, integration and interoperability. To make the discovery and use of all these semantic artefacts easier, LifeWatch ERIC developed EcoPortal. The platform acts as an access point for semantic artefacts, allowing users to maximise the efficiency of search, annotation and other scientific data management processes as well as to perform specific tasks. All of these functions are seamlessly integrated within a fully compliant semantic web infrastructure.

2. EcoPortal overview

2.1 Homepage

The EcoPortal homepage (Figure 1) consists of:

- a header containing the logo on the left and the [navigation bar](#) on the right;
- a [search bar](#) to quickly search for terms or for semantic artefacts by entering a free text. The autocomplete functionality will support the quest by suggesting terms or semantic artefacts available within the portal;
- some statistical information, provided as bubbles cloud, indicating the number of visits for the EcoPortal semantic artefacts;
- EcoPortal metrics such as number of published resources, number of classes, etc.;

- a [recommender](#) and [annotator](#) box;
- the [O'FAIR tool](#) with the average FAIR score of the semantic artefacts published within EcoPortal;
- a news box displaying the most recent posts of LifeWatch ERIC;
- the links to other OntoPortal instances;
- the EcoPortal collaborations and support;
- a [footer](#) containing additional supporting information and documentation.

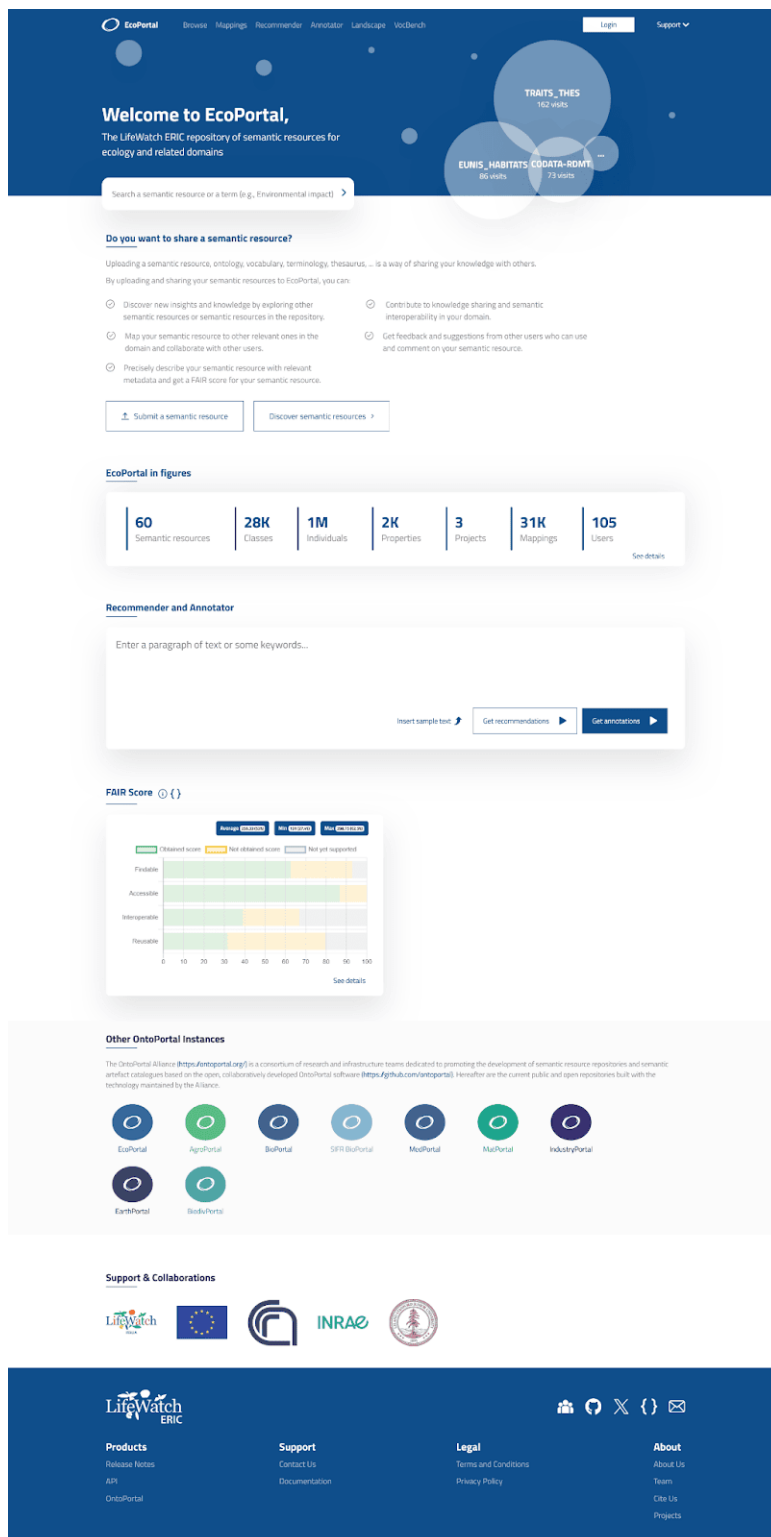
2.1.1 Navigation bar

The navigation bar at the top of the homepage contains eighth menu items for navigating within EcoPortal:

- **Browse** - to search semantic artefacts.
- **Mappings** - to explore mappings and upload them.
- **Recommender** - to get recommendations from semantic artefacts' terms.
- **Annotator** - to get annotations with semantic artefacts' terms.
- **Landscape** - to visualise data retrieved from analyses on semantic artefacts metadata.
- **VocBench** - to create or edit a semantic artefact.
- **Login** - to access the login/registration form. Once logged in, username will be displayed here and it will show the access to [account setting](#) and "logout" button.
- **Support:**
 - **Send Feedback** - to request assistance with questions, bug reports, feedback, and so on through an easy-to-use contact form.
 - **Documentation** - to access the EcoPortal documentation.
 - **Cite us** - to find the citation and read the publications related to EcoPortal.
 - **Release Notes** - to read the last release notes.
 - **Groups and Categories** - to find more information about groups and categories available in EcoPortal.

2.1.2 Account settings

In the navigation bar, the Account Settings page can be accessed to view all the account details: personal information, submitted semantic artefacts, mailing list subscription, customisation options and the personal API Key (Figure 2).



EcoPortal Browse Mappings Recommender Annotator Landscape VocabBench Login Support

Welcome to EcoPortal,
The LifeWatch ERIC repository of semantic resources for ecology and related domains

Search a semantic resource or a term (e.g., Environmental impact) >

Do you want to share a semantic resource?
Uploading a semantic resource, ontology, vocabulary, terminology, thesaurus, ... is a way of sharing your knowledge with others. By uploading and sharing your semantic resources to EcoPortal, you can:

- Discover new insights and knowledge by exploring other semantic resources or semantic resources in the repository.
- Map your semantic resource to other relevant ones in the domain and collaborate with other users.
- Precisely describe your semantic resource with relevant metadata and get a FAIR score for your semantic resource.
- Contribute to knowledge sharing and semantic interoperability in your domain.
- Get feedback and suggestions from other users who can use and comment on your semantic resource.

Submit a semantic resource Discover semantic resources >

EcoPortal in figures

60 Semantic resources	28K Classes	1M Individuals	2K Properties	3 Projects	31K Mappings	105 Users
--------------------------	----------------	-------------------	------------------	---------------	-----------------	--------------

See details

Recommender and Annotator

Enter a paragraph of text or some keywords...

Insert sample text > Get recommendations > Get annotations >

FAIR Score { }

Legend: Obtained score (green), Not obtained score (yellow), Not yet supported (grey)

FAIR Principle	Obtained score	Not obtained score	Not yet supported
Findable	100%	0%	0%
Accessible	100%	0%	0%
Interoperable	100%	0%	0%
Reusable	100%	0%	0%

See details

Other OntoPortal Instances

The OntoPortal Alliance (<https://ontoportal.org/>) is a consortium of research and infrastructure teams dedicated to promoting the development of semantic resource repositories and semantic web/rdf catalogues based on the open, collaboratively developed OntoPortal software (<https://github.com/ontoportal>). Hereafter are the current public and open repositories built with the technology maintained by the Alliance.

- EcoPortal
- AgroPortal
- BioPortal
- SFR BioPortal
- MedPortal
- MusPortal
- IndustryPortal
- EarthPortal
- BiodidPortal

Support & Collaborations

LifeWatch ERIC European Union INRAE

LifeWatch ERIC

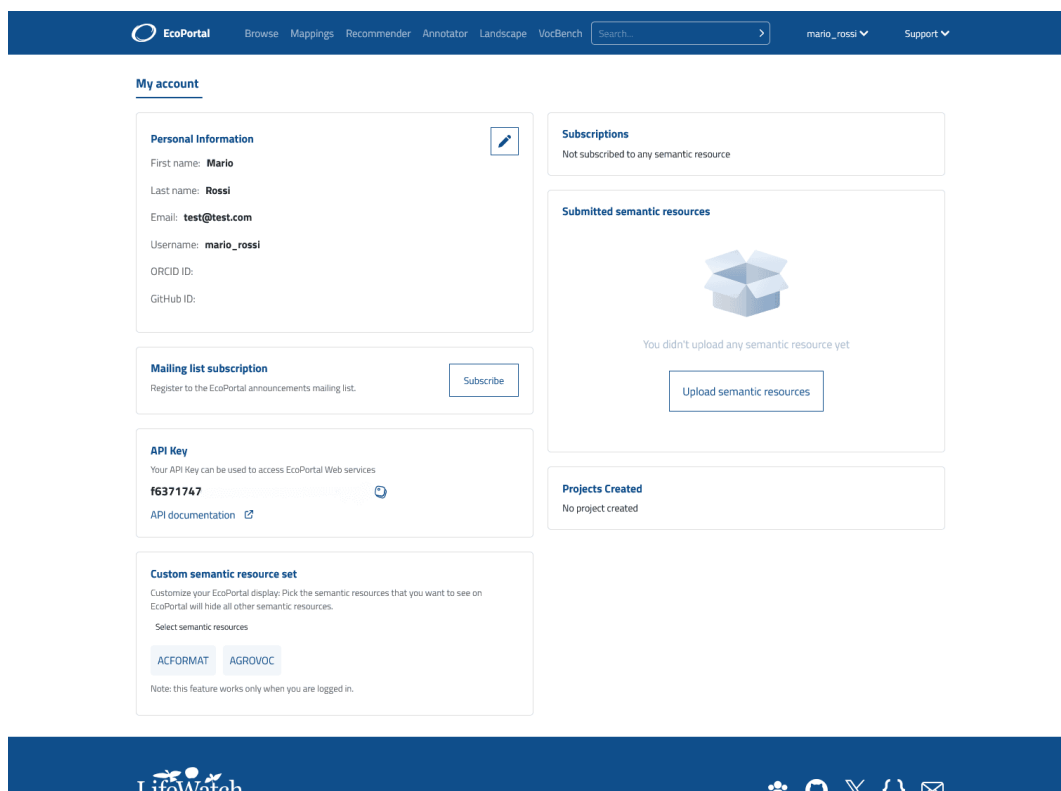
Products
Release Notes
API
OntoPortal

Support
Contact Us
Documentation

Legal
Terms and Conditions
Privacy Policy

About
About Us
Team
Cite Us
Projects

Figure 1: EcoPortal homepage



EcoPortal Browse Mappings Recommender Annotator Landscape VocBench Search... mario_rossi Support

My account

Personal Information [Edit]

First name: **Mario**
Last name: **Rossi**
Email: **test@test.com**
Username: **mario_rossi**
ORCID ID:
GitHub ID:

Subscriptions
Not subscribed to any semantic resource

Submitted semantic resources

You didn't upload any semantic resource yet

[Upload semantic resources](#)

Mailing list subscription
Register to the EcoPortal announcements mailing list. [Subscribe](#)

API Key
Your API Key can be used to access EcoPortal Web services
f6371747 [Copy]
[API documentation](#)

Projects Created
No project created

Custom semantic resource set
Customize your EcoPortal display: Pick the semantic resources that you want to see on EcoPortal will hide all other semantic resources.
Select semantic resources
[ACFORMAT](#) [AGROVOC](#)
Note: this feature works only when you are logged in.

LifeWatch [Icons]

Figure 2: Account settings page

2.1.3 O'FAIR tool

The Ontology FAIRness Evaluator (O'FAIR) tool is a web service that performs an automatic FAIRness evaluation of semantic artefacts by analysing metadata fields associated with particular questions corresponding to each FAIR principle and sub-principle.

The tool automatically executes the tests and evaluates how the semantic artefacts respond to the questions, providing a global normalised FAIR score associated with some basic statistics. To better understand which are the features of the tool and how the FAIRness evaluation is executed for each semantic artefact, please go [here](#). To learn more about the tool please visit the [GitHub page](#).

2.1.4 Footer

The footer contains all the EcoPortal support links, divided into five sections:

- **Products:**
 - Release Notes

- API
- OntoPortal
- **Support:**
 - Contact Us
 - Documentation
- **Legal:**
 - Terms & Conditions
 - Privacy Policy
- **About:**
 - About us
 - Team
 - Cite Us
 - Projects

2.2 EcoPortal tools

EcoPortal provides different tools to access, use and manage semantic artefacts including:

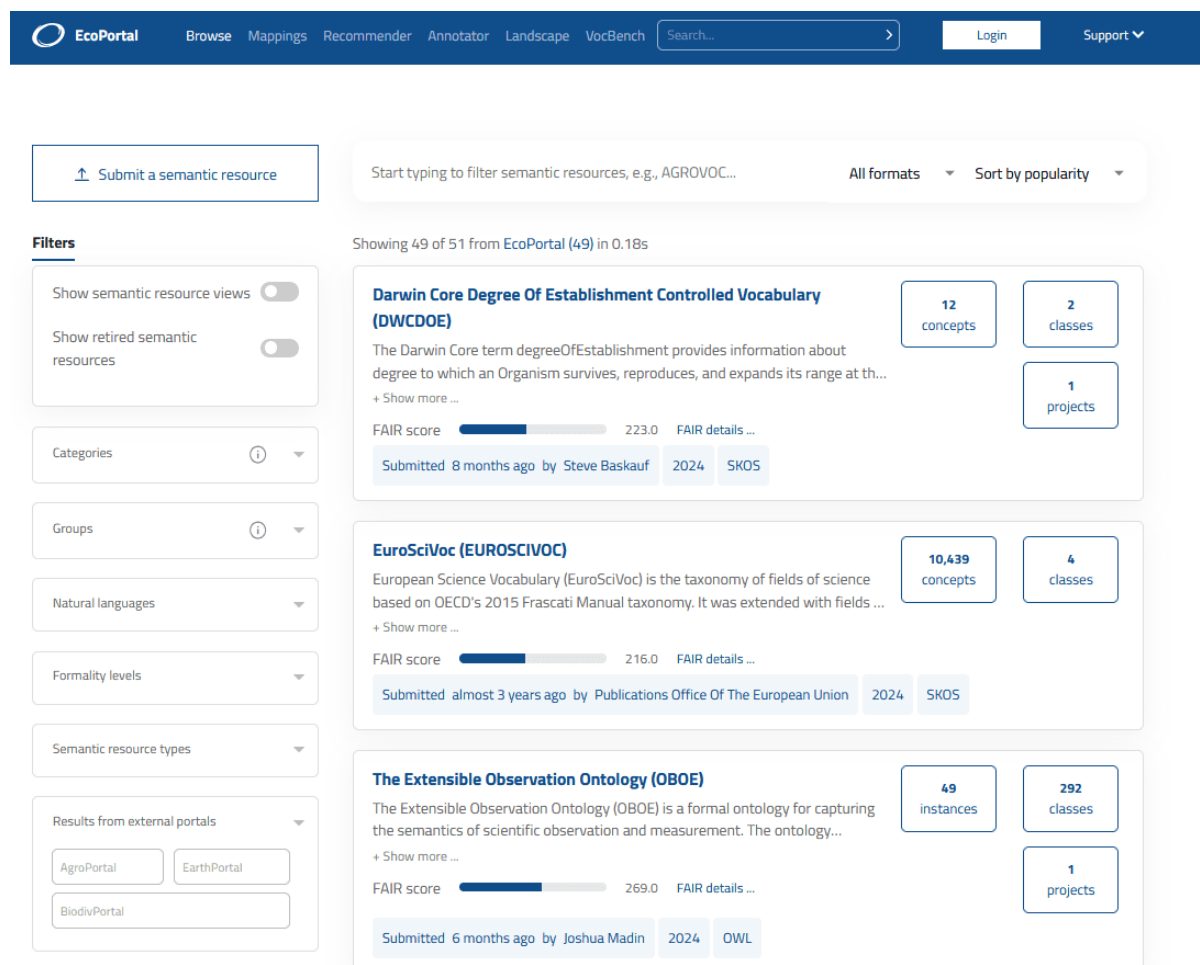
- [browse](#) the semantic artefacts;
- [search](#) for specific class or concepts;
- [browse mappings](#) among terms in different semantic artefacts;
- [receive recommendations](#) on which semantic artefacts are most relevant for a given corpus;
- [annotate text](#) with terms from semantic artefacts;
- [create and edit semantic artefacts](#) with VocBench Editor.

All information available through the EcoPortal website is also available through the [REST API documentation](#).

2.2.1 Browse

The Browse tool allows users to search for semantic artefacts. When opening the browser page (Figure 3), there are two options:

1. scroll the full list of semantic artefacts and select one to see more detailed information (Browse all);
2. type the semantic artefact name or abbreviation in the Search bar.



The screenshot shows the EcoPortal interface. At the top is a navigation bar with links: EcoPortal, Browse, Mappings, Recommender, Annotator, Landscape, VocBench, a search bar, Login, and Support. Below the navigation bar, there's a section for submitting a semantic resource. The main area is titled 'Browse' and shows a search bar with the text 'Start typing to filter semantic resources, e.g., AGROVOC...'. To the right of the search bar are dropdowns for 'All formats' and 'Sort by popularity'. Below the search bar, it says 'Showing 49 of 51 from EcoPortal (49) in 0.18s'. On the left side, there are several filter sections: 'Filters' with toggle switches for 'Show semantic resource views' and 'Show retired semantic resources'; 'Categories' with a dropdown; 'Groups' with a dropdown; 'Natural languages' with a dropdown; 'Formality levels' with a dropdown; 'Semantic resource types' with a dropdown; and 'Results from external portals' with buttons for 'AgroPortal', 'EarthPortal', and 'BiodivPortal'. The main results area displays three semantic resources: 1. 'Darwin Core Degree Of Establishment Controlled Vocabulary (DWCDOE)' with 12 concepts, 2 classes, and 1 project. 2. 'EuroSciVoc (EUROSCIVOC)' with 10,439 concepts and 4 classes. 3. 'The Extensible Observation Ontology (OBOE)' with 49 instances, 292 classes, and 1 project. Each result includes a FAIR score bar, a 'Show more...' link, and submission details.

Figure 3: Browse page

In both cases, the search can be further refined by using the filters provided on the left side of the page (*Show semantic resource views*, *Show retired semantic resources*, *Categories*, *Groups*, *Natural languages*, *Formality levels*, *Semantic resource types*). Upon selecting the filter, the search will be updated to show results which contain the selected filter. Repeat the process to add additional filters.

Moreover, using the tabs on the right side of the search bar, the list of semantic artefacts can be sorted by choosing the type of format (*SKOS*, *OWL*, *UMLS*, *OBO*) or other desired criteria:

- *Name*: alphabetical order;
- *Classes count*: number of classes;
- *Instances/concepts count*: number of instances or concepts;
- *Submitted date*: date of last submission;

- *Creation date*: date of publication;
- *FAIR score*: O'FAIRe tool score;
- *Popularity*: number of visits to the semantic artefact;
- *Notes*: number of notes issued for the semantic artefacts;
- *Projects*: number of projects where semantic artefacts are used.

To access semantic artefacts through the API, see our [Resources REST API documentation](#).

In the browse page, the "Submit new semantic artefact" button is also available on the upper-left part of the page, but users **need to be logged in** to perform this operation.

For more information about publishing a new semantic artefact in EcoPortal, please visit the [guide](#).

2.2.2 Search

The [Search](#) option allows users to find a term across semantic artefacts by clicking on the arrow pointing to the right in the home page search box and entering free text in the "Enter a term" bar. The system looks for matches in the term name, synonyms, term IDs, and in property values.

The advanced search options can be used to add (Figure 4):

- *Property values*: named association between two entities. Examples are "definition" (a relationship between a class and some text) and "part of" (a relationship between two classes).
- *Obsolete classes*: a class that the authors of the semantic artefact have flagged as being obsolete. These classes are often left in semantic artefacts so that existing systems that depend on them will continue to function.
- *Semantic resource views*: search only ontological views.

Users can also narrow the search by selecting "exact matches" and/or "classes with definitions" or by selecting one or more categories and/or semantic artefacts using the "Categories" and "Semantic resources" fields. In addition, it is possible to perform a broader search by selecting one or more federated OntoPortal instances, namely AgroPortal, BiodivPortal and EarthPortal, from the section "Results from external portals".

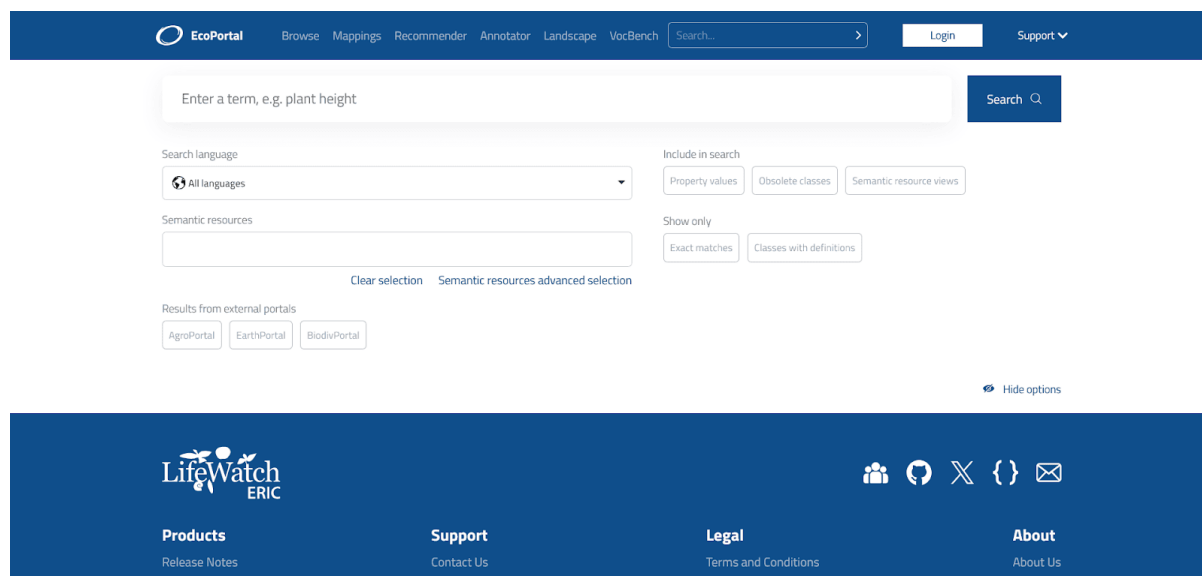


Figure 4: The search tool

To allow better performances, the search result is limited to the top 100 matches.

The search result within EcoPortal will display the matched terms and their associated semantic artefact. For each search result it is also shown:

- *details*: a window with the associated properties of the searched term;
- *visualize*: the graph with the nodes of the terms and the path to the root;
- *<number> more from this semantic resource*: other similar terms within the semantic artefact are shown.

If a federated search is performed, for each search result it is shown:

- *<number> more from this semantic resource*;
- the badge of the OntoPortal instance in which the result can be found.

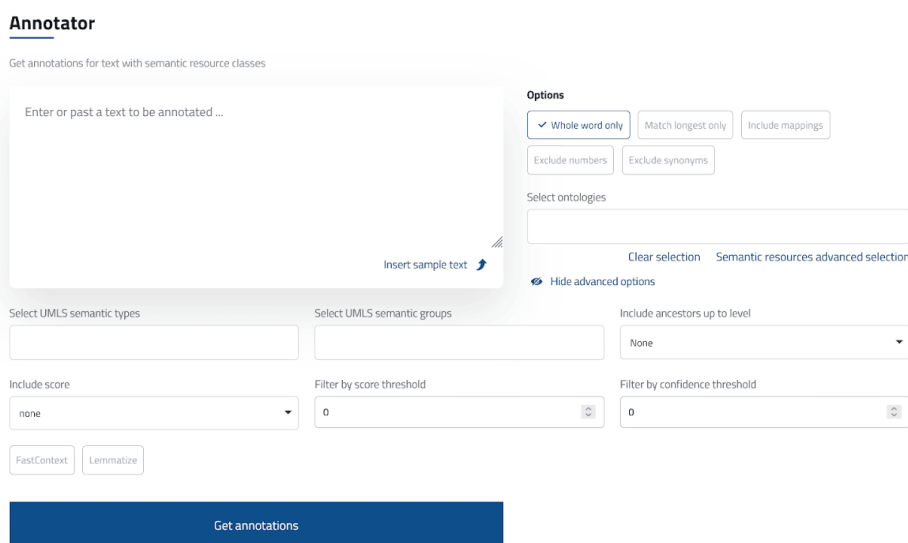
Search can be done using the API directly. To learn more, see our [Search REST API documentation](#).

2.2.3 Annotator

The Annotator tool enables the generation of annotations by entering or pasting free text in the box and clicking on the "Get Annotations" button (Figure 5).

The system matches words in the text to terms from the semantic artefacts available in the portal by doing an exact string comparison (a "direct" match) between the text and names, synonyms, and IDs of terms within the semantic artefacts. The set of matches can be expanded by including matches from mapped terms and from hierarchical

expansion. The system performs the hierarchical expansion on the superclass ("is-a") relationship for most semantic artefacts, including OWL and UMLS RFF. For OBO semantic artefacts the hierarchical expansion also includes the "part-of" relationship. The "number of levels" can be used to control the number of levels up the hierarchy for which the system will return terms for a given match. For more details about the Annotator tool, please read [Shah et al., 2009](#). To generate annotations from the API, see the [REST API documentation](#).



The screenshot shows the 'Annotator' web interface. At the top, it says 'Get annotations for text with semantic resource classes'. Below this is a large text input area with the placeholder 'Enter or paste a text to be annotated ...' and an 'Insert sample text' button. To the right of the input area is an 'Options' panel with several settings: 'Whole word only' (checked), 'Match longest only', 'Include mappings', 'Exclude numbers', and 'Exclude synonyms'. Below these are 'Select ontologies' and 'Clear selection' buttons. Further down is a 'Hide advanced options' link. Below the input area are three dropdown menus: 'Select UMLS semantic types', 'Select UMLS semantic groups', and 'Include ancestors up to level' (set to 'None'). Below these are three more dropdowns: 'Include score' (set to 'none'), 'Filter by score threshold' (set to '0'), and 'Filter by confidence threshold' (set to '0'). At the bottom left are 'FastContext' and 'Lemmatize' buttons. A large blue 'Get annotations' button is at the bottom center.

Figure 5: The annotator tool

2.2.4 Mappings

The Mappings tool enables the association of two or more terms in different semantic artefacts, typically indicating a degree of similarity between them. The author of the mapping defines its relation; in addition, although not required, it is usually recommended for a mapping to be bi-directional.

To upload mappings, use the "Upload mappings" button. A valid file format example is available. You can either select the file or drag and drop it into the box.

Mappings can be searched in two ways:

1. Find mappings of classes/concepts from the "**Bubble view**". Each bubble is a semantic artefact and its size represents the number of mappings available (Figure 6). After selecting two semantic artefacts, a box will appear displaying the mappings between those two artefacts (Figure 7). Select the desired class/concept to view a table with the same information as shown [here](#).

2. Browse mappings between classes/concepts in different semantic artefacts from the "**Table view**" by selecting the semantic artefact from the drop-down menu. The results table will show all the semantic artefacts to which the selected resource is mapped with and the number of existing mappings (minimum one) (Figure 8). Select a semantic artefact from this table to browse the mappings between the two semantic artefacts.

To access mappings from the API, please see our [Mappings REST API documentation](#).

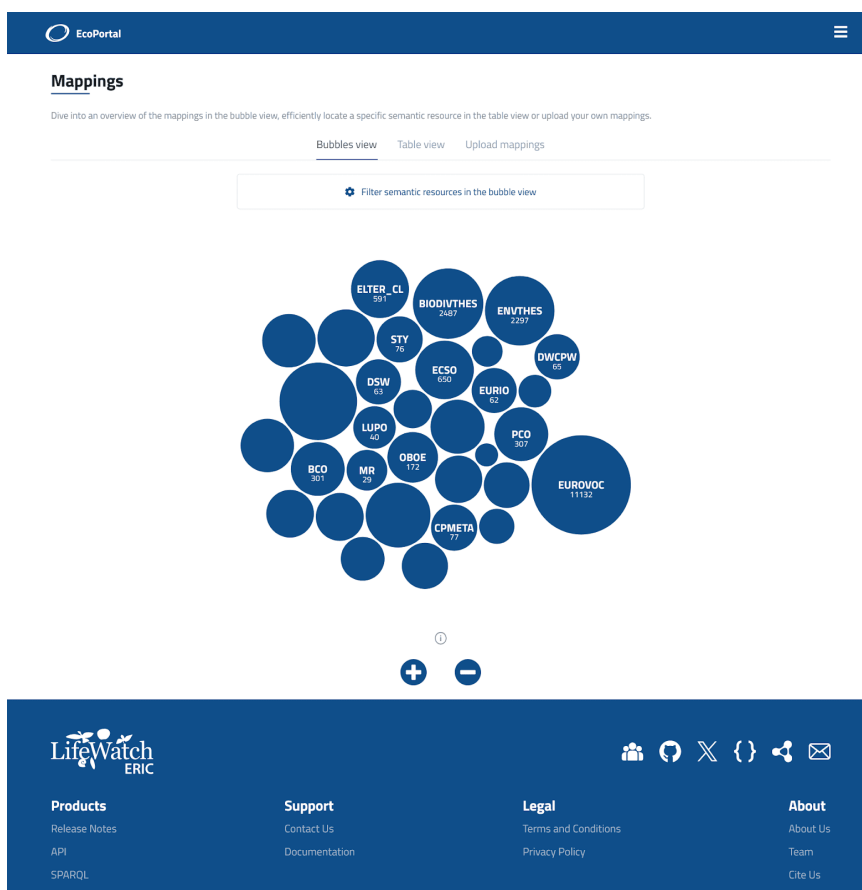
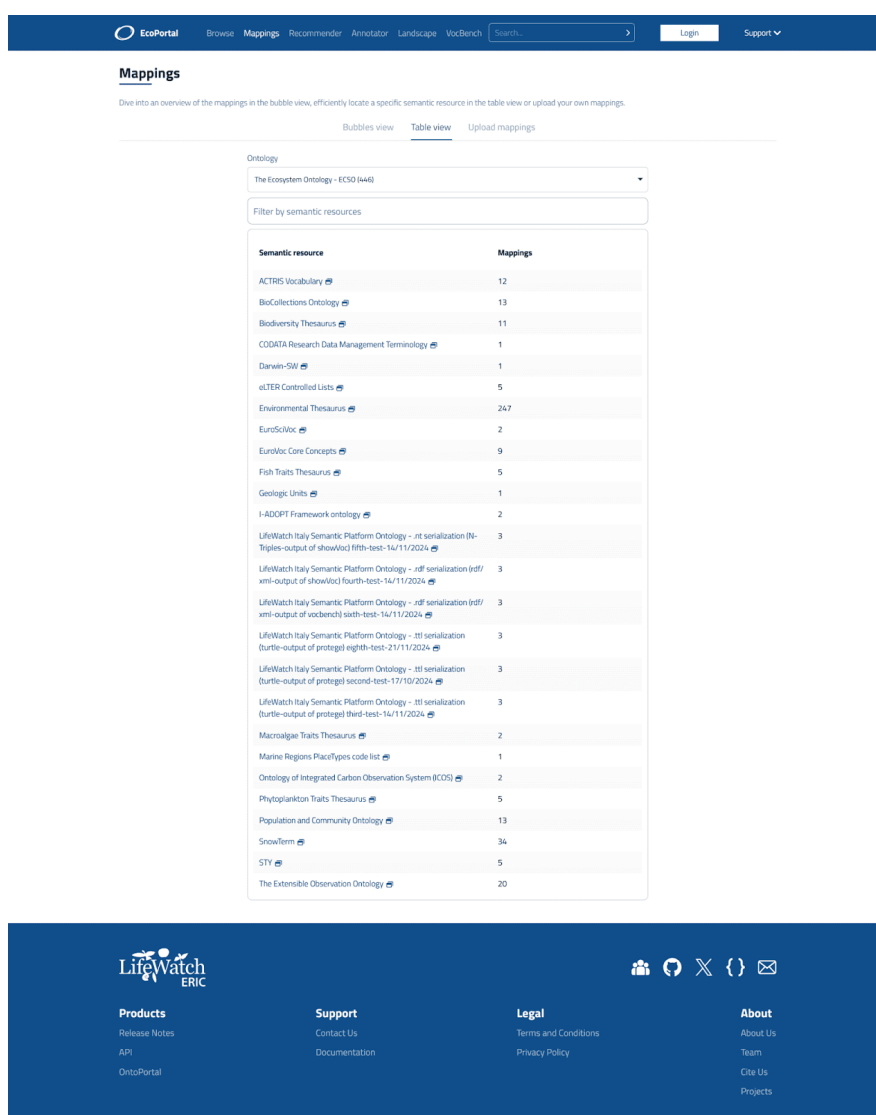


Figure 6: Browse mappings between classes in different semantic artefacts

ECVO	OBCE	Source
VolumetricRate >	VolumetricRate >	SAME_URI
Protocol >	Protocol >	SAME_URI
mass density >	MassDensity >	LOOM
AreaMassDensity >	AreaMassDensity >	SAME_URI
Illuminance >	Illuminance >	LOOM
resistivity >	Resistivity >	LOOM
Length >	Length >	LOOM
MeasuredCharacteristic >	MeasuredCharacteristic >	SAME_URI

Figure 7: Mappings of a class/concept



Mappings

Dive into an overview of the mappings in the bubble view, efficiently locate a specific semantic resource in the table view or upload your own mappings.

Bubbles view **Table view** Upload mappings

Ontology
The Ecosystem Ontology - ECVO (446)

Filter by semantic resources

Semantic resource	Mappings
ACTRIS Vocabulary	12
BioCollections Ontology	13
Biodiversity Thesaurus	11
CODATA Research Data Management Terminology	1
Darwin-SW	1
eLTER Controlled Lists	5
Environmental Thesaurus	247
EuroSciVoc	2
EuroVoc Core Concepts	9
Fish Traits Thesaurus	5
Geologic Units	1
I-ADOPT Framework ontology	2
LifeWatch Italy Semantic Platform Ontology - nt serialization (N-Triples-output of showVoc) fifth-test-14/11/2024	3
LifeWatch Italy Semantic Platform Ontology - rdf serialization (rdf/xml-output of showVoc) fourth-test-14/11/2024	3
LifeWatch Italy Semantic Platform Ontology - rdf serialization (rdf/xml-output of vocbench) sixth-test-14/11/2024	3
LifeWatch Italy Semantic Platform Ontology - ttl serialization (turtle-output of protogol eighth-test-21/11/2024)	3
LifeWatch Italy Semantic Platform Ontology - ttl serialization (turtle-output of protogol second-test-17/10/2024)	3
LifeWatch Italy Semantic Platform Ontology - ttl serialization (turtle-output of protogol third-test-14/11/2024)	3
Macroalgae Traits Thesaurus	2
Marine Regions Placenames code list	1
Ontology of Integrated Carbon Observation System (ICOS)	2
Phytoplankton Traits Thesaurus	5
Population and Community Ontology	13
SnowTerm	34
STY	5
The Extensible Observation Ontology	20

Products
Release Notes
API
OntoPortal

Support
Contact Us
Documentation

Legal
Terms and Conditions
Privacy Policy

About
About Us
Team
Cite Us
Projects

Figure 8: Find mappings of a class/concept

Mapping Sources

Mappings are generated by different sources in two different ways: automatic mappings and user generated mappings. Automatic mappings are generated by the portal from:

- **LOOM**. This algorithm automatically computes mapping based on close lexical matches between preferred names of terms or a preferred name and a synonym. The lexical match is performed if white-spaces and punctuation are removed from the labels. All labels with less than 3 characters are excluded. These mappings are represented by the mapping relationship "skos:closeMatch".
- **SameURI**. Generated between two concepts with the same URI.
- **CUI**. Generated between two concepts that have the same CUI (Concept Unique Identifier).

User-generated mappings are uploaded and stored within the portal as **RESTBackupMapping**, namely a mapping added by a user using the REST API (or the UI, which is calling the API to create it).

In addition to these two, **skos:exactMatch** and **skos:closeMatch** mappings, available in the semantic artefacts files, are extracted and displayed in the results table. The "type" is indicated as SKOS. **skos:relatedMatch** mappings present in the semantic artefacts files **are not** extracted and shown in the table, but they are visible in the "Detail" tab of the class or concept.

For more information on how the mappings are handled inside the triple store, please refer to [this documentation](#).

2.2.5 Recommender

The Recommender tool allows users to find the most relevant semantic artefacts based on text excerpts or list of keywords (Figure 9). The ranking algorithm used by the Recommender evaluates the appropriateness of each semantic artefact available in the portal using a combination of four criteria:

- *Coverage*: to what extent does the semantic artefact represent the input? The Recommender invokes the Annotator service to obtain all the annotations of the input and then uses these annotations to compute a coverage score for each semantic artefact found.
- *Acceptance*: how well-known and trusted by the ecological community is the semantic artefact? The number of visits to the semantic artefact page in EcoPortal are used to compute an acceptance score for each semantic artefact found.

- *Detail of knowledge*: what is the level of detail provided by the semantic artefact for the input data? It is computed using the number of definitions, synonyms and properties of the semantic artefact classes/concepts that cover the input data.
- *Specialisation*: how specialised is the semantic artefact with respect to the domain of input data? This is calculated using the number and type of annotations performed with the semantic artefact and the position of each annotated class/concept in the semantic artefact hierarchy. The result is normalised by the size of the semantic artefact, in order to identify small semantic artefacts that are specialised to the input data.

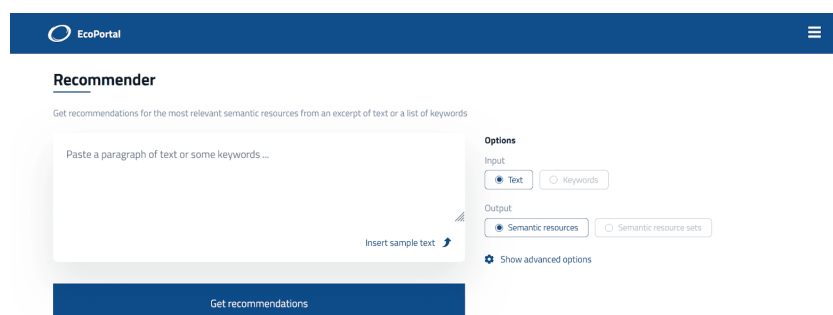


Figure 9: The recommender tool

For each of these four criteria, a score is computed. Then, the scores obtained are weighted and aggregated into a final score for each semantic artefact found. The default weights used to aggregate the scores can be changed in the advanced options section.

For more details about the Recommender tool and to cite it, please read the related publication of [Martínez-Romero et al., 2017](#).

To retrieve semantic artefact recommendations from the API, please see our [Recommender REST API documentation](#).

2.2.6 VocBench

VocBench is a web-based, multilingual, collaborative development platform, designed to meet the needs of semantic web and linked data environments. It allows to manage [OWL](#) ontologies, [SKOS\(/XL\)](#) thesauri, [Ontolex-lemon](#) lexicons and generic RDF datasets. VocBench business and data access layers are realised by [Semantic Turkey](#), an open-source platform for Knowledge Acquisition and Management realised by the [ART Research Group](#) at the [University of Rome Tor Vergata](#). EcoPortal uses VocBench as an

editor tool for managing semantic artefacts with a customised version (v. 13).

User Interface

The user interface detailing the resource being edited on VocBench contains three sections (Figure 10):

- the Global Data Management
- the menu
- the working space that changes based on the tabs selected

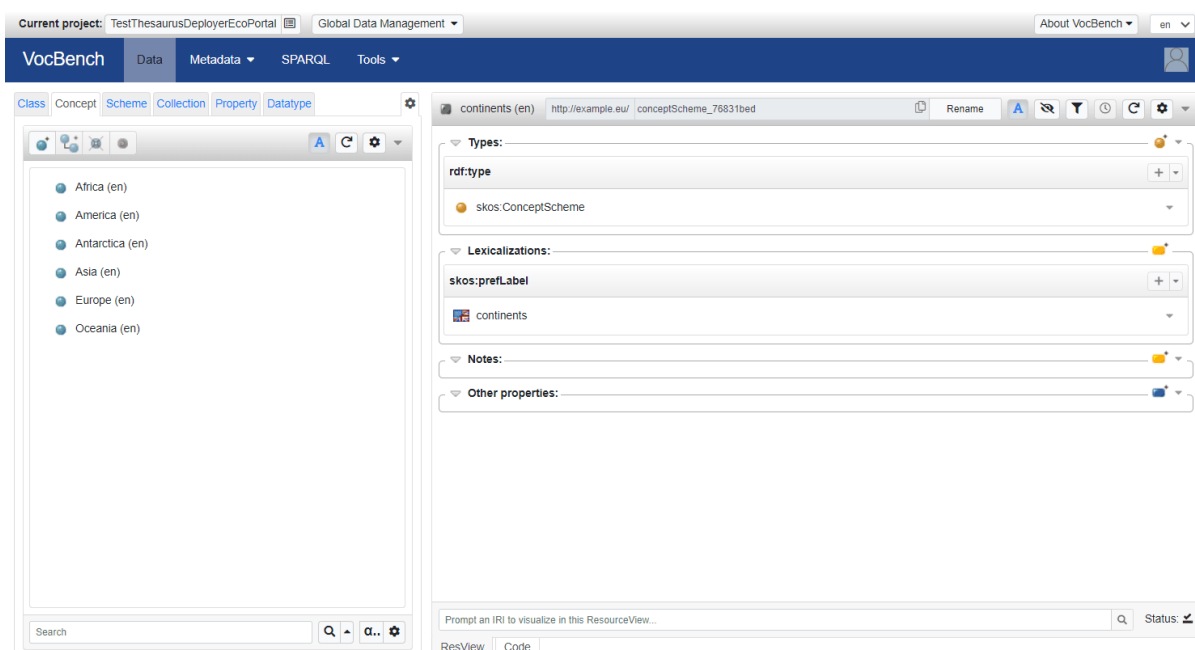


Figure 10: VocBench UI

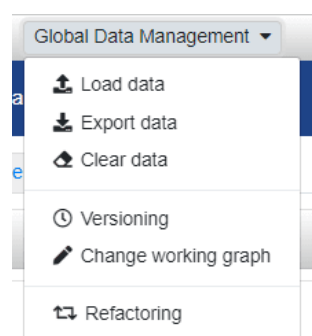


Figure 11: Global Data Management menu

In the [Global Data Management](#) (Figure 10.a) users, depending on their role (for more information please read [this](#)), can access to different functionalities (Figure 11):

- *Load data* to upload data that has to be maintained within the project.
- *Export data* to download or deploy the resource in EcoPortal.
- *Clear data* to clear completely the project repository.
- *Versioning* to create time-stamped data dumps of the dataset, that can later be inspected through the same project.
- *Change working graph* to manage variants of graphs.
- *Refactoring* to perform massive refactoring of the loaded data.

In the menu section (Figure 10.b), users can access:

- [Data](#) which is the main portion in which all the editing of the resource occurs;
- [Metadata](#) with a drop down menu which has three entries;
 - [Namespaces and Import](#) page, allows users to set prefix-namespace mappings, to owl:import ontology vocabularies and to edit the ontology mirror, a local mirror of ontologies stored within VocBench;
 - [Metadata Vocabularies](#) page, allows for the specification of metadata according to different existing metadata vocabularies, such as VOID, LIME, DCAT, ADMS, etc.;
 - [Metadata Registry](#) page, allows authorised users to define a system-wide catalogue of known remote datasets, which are described by means of a combination of (metadata) vocabularies;
- [SPARQL](#) enable users to access to the SPARQL query editor of the resource;
- [Tools](#) with a drop down menu showing some tools that expand the VocBench core capabilities.

The official VocBench documentation can be found at the following [link](#). For a detailed description of the VocBench version developed within EcoPortal, please refer to the following [section](#).

Note: At the moment, users must register separately to access VocBench and other EcoPortal services.

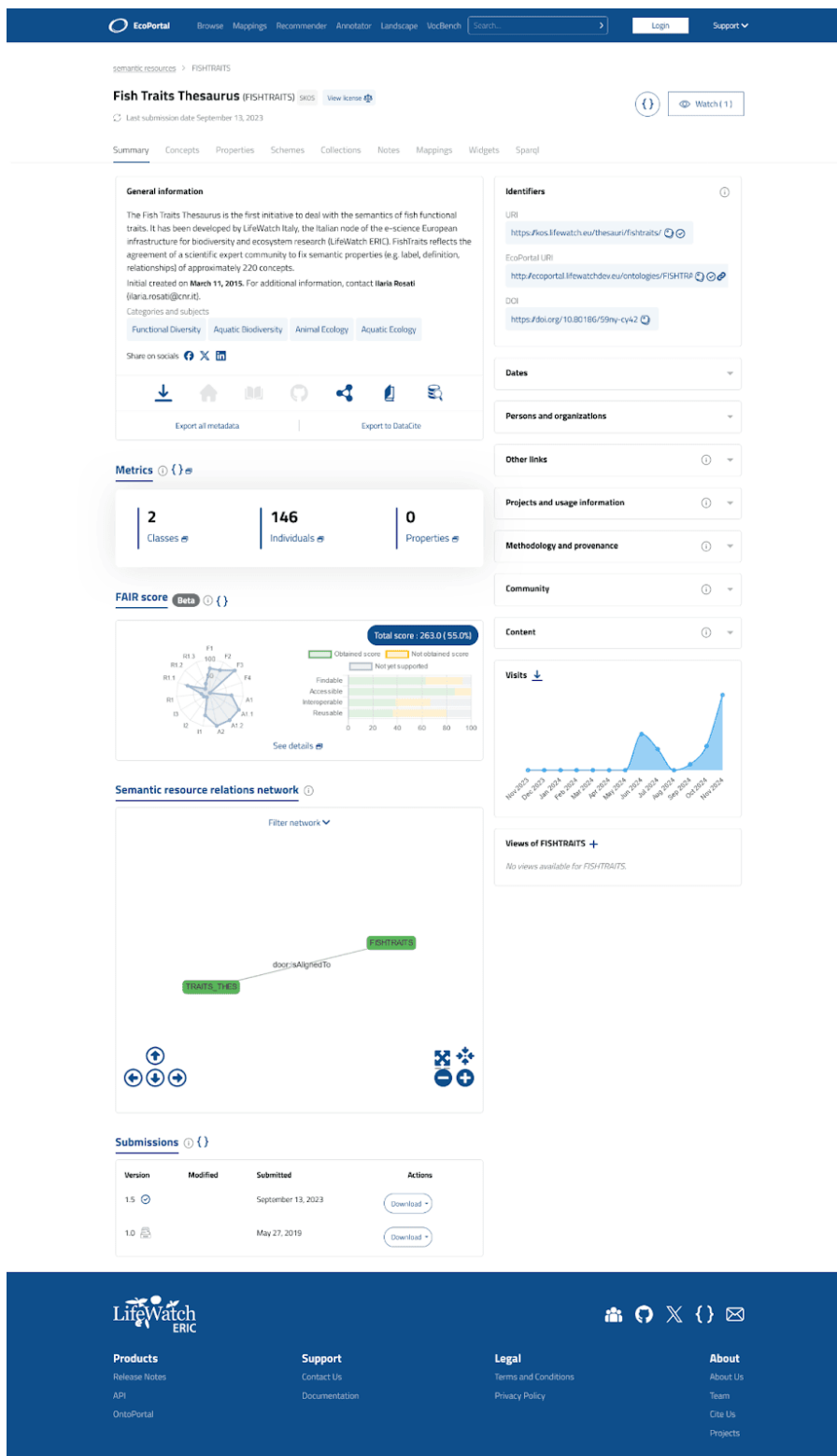
2.3 Semantic artefact detail page

Once the selected semantic artefact has been opened, the summary page shows the different sections containing its [description](#) (Figure 12). On the left are displayed:

- a [menu](#) with different tabs;
- general information box;
- links to the homepage, associated publications and documentation, SPARQL endpoint, download buttons of the semantic artefact and links to the platform's social media;
- [export metadata buttons](#) for metadata in different formats (JSON DATACITE, JSON MOD, N-Triple, JSON-LD, RDF/XML and CSV);
- [metrics](#) table;
- [O'FAIR tool](#) with the score for the selected semantic artefact;
- Semantic artefacts relations network, which is shown only when metadata related to the semantic artefact relations (door:isAlignedTo, dct:hasPart, owl:imports, ect.) are filled;
- [submissions](#).

On the right are displayed:

- link to the REST API JSON entry: to access the JSON entry users must be logged in;
- watch button, to stay updated on the status of the semantic artefact;
- other metadata information including identifiers, dates, projects and usage information, methodology and provenance, community and more content information.
- a downloadable visits graph showing the monthly number of visualisations of the semantic artefact;
- [views](#) of the semantic artefact.



Fish Traits Thesaurus (FISHTRAITS) SKOS View license

Last submission date September 13, 2023

General information

The Fish Traits Thesaurus is the first initiative to deal with the semantics of fish functional traits. It has been developed by LifeWatch Italy, the Italian node of the e-science European infrastructure for biodiversity and ecosystem research (LifeWatch ERIC). FishTraits reflects the agreement of a scientific expert community to fix semantic properties (e.g. label, definition, relationships) of approximately 220 concepts.

Initial created on March 11, 2015. For additional information, contact ilaria.rosati@icnr.it.

Categories and subjects

Functional Diversity Aquatic Biodiversity Animal Ecology Aquatic Ecology

Share on social: [Facebook](#) [Twitter](#) [LinkedIn](#)

Export all metadata | Export to DataCite

Metrics

2 Classes | 146 Individuals | 0 Properties

FAIR score (Beta)

Total score: 263.0 (55.0%)

Obtained score | Not obtained score

Findable | Reusable | Interoperable | Reusable

Semantic resource relations network

Filter network

fishtraits:traits

Submissions

Version	Modified	Submitted	Actions
1.5		September 13, 2023	Download
1.0		May 27, 2019	Download

Identifiers

URI: <https://icos.lifewatch.eu/thesauri/fishtraits/>

EcoPortal URI: <http://ecoportal.lifewatchdev.eu/ontologies/FISHTRAP>

DOI: <https://doi.org/10.80186/59my-cy42>

Visits

Views of FISHTRAITS: No views available for FISHTRAITS.

Figure 12: The summary page of semantic artefacts

2.3.1 Summary

The information contained in this page is consistent with the information provided during the submission of the semantic artefact, which adheres to the MOD 2.0. For further insight into the metadata schema, click [here](#).

■ Metadata export

On the left side of the summary page, below the general information box, there are two buttons for downloading metadata of different formats:

- Export all metadata:
 - **CSV** button provides the semantic artefact metadata in CSV format;
 - **N-Triples** button provides the semantic artefact metadata in N-Triples format;
 - **JSON-LD** button provides the semantic artefact metadata in JSON-LD format;
 - **RDF/XML** button provides the semantic artefact metadata in RDF/XML format.
- Export to DataCite:
 - **JSON** button extracts the DataCite metadata in JSON format.

■ FAIR Score

The FAIR score displayed as "Total score" is specific for the semantic artefact. By selecting the details button, as illustrated in Figure 13, users can explore sets of questions associated with each FAIR Sub-principle, along with their corresponding credits and utilised metadata properties (refer to Figure 14). The system then automatically computes the score.

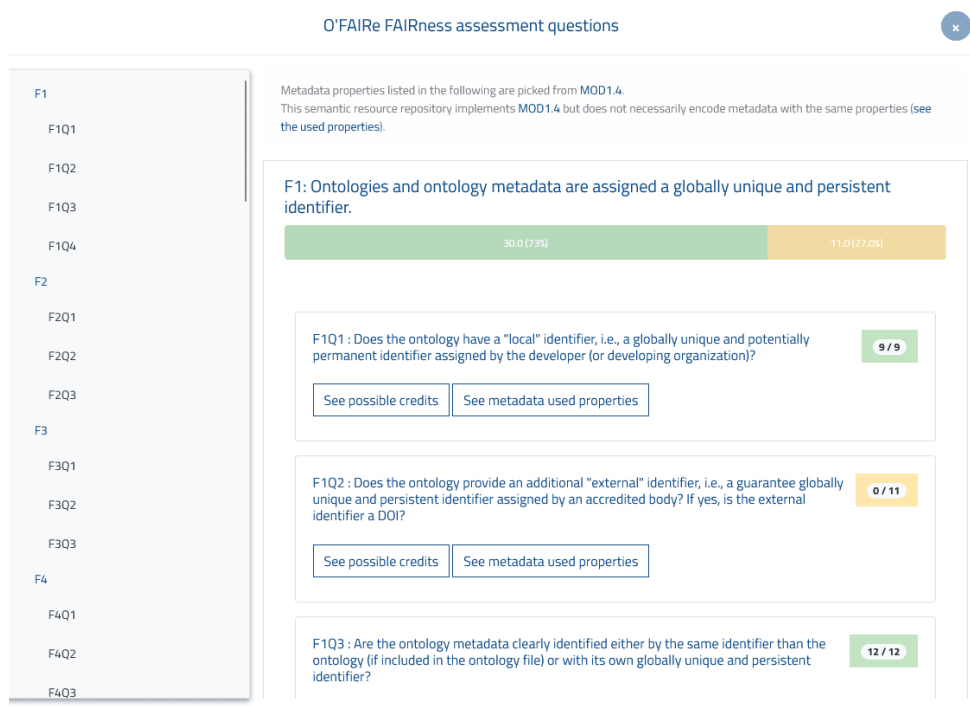


Figure 13: Details view of the O'FAIRe tool

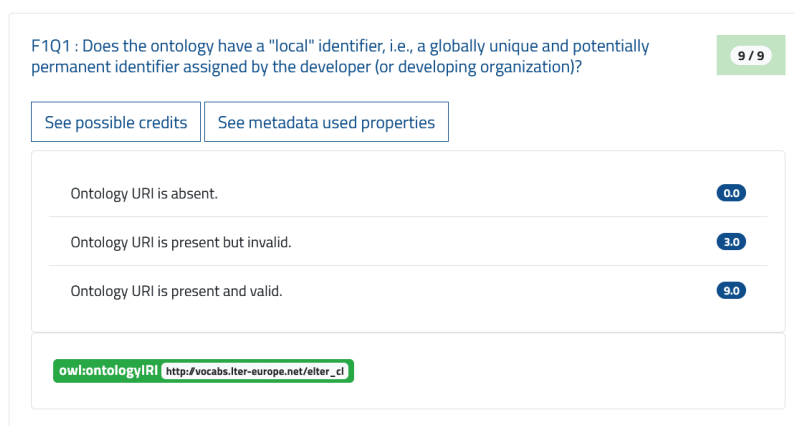


Figure 14: Credits and metadata used properties of a question

■ Metrics

This section describes the metrics that EcoPortal computes for each semantic artefact (Figure 15). Metrics start to be computed when a semantic artefact is uploaded and they are part of the [EcoPortal Metadata Schema](#). By clicking on the window icon next to "Classes", "Individuals" and "Properties", users can access the graph showing the evolution of the number of classes/individuals/properties of the selected semantic artefact.

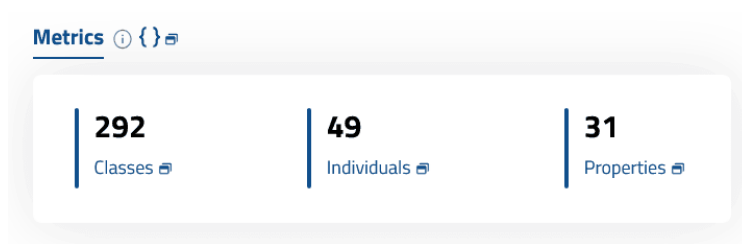


Figure 15: Metrics area of the summary page of a semantic artefact

Note: some metrics are meaningful only for semantic artefacts in a specific representation language (e.g., there are no individuals to count in the ontologies in OBO format; there is no SKOS concepts count).

The statistical metrics include:

- *Number Of Classes*: the number of named (not anonymous) classes in the semantic artefact;
- *Number Of Individuals*: the number of individuals in the semantic artefact;
- *Number Of Properties*: the number of properties or slots in the semantic artefact;

For more details about the metrics generation, please read [this](#). EcoPortal users will be able to access semantic artefact metrics in two ways:

- through the EcoPortal user interface, in the metrics area of the summary page (see example [here](#));
- through a dedicated REST service that returns a JSON (see example below).

The following REST service can be used:

https://data.ecoportal.lifewatch.eu/ontologies/ELTER_CL/metrics?apikey=YourAPIKey

The service returns a JSON that contains the version id for the semantic artefact and the values for the metrics:

```
{
  'id': 'https://data.ecoportal.lifewatch.eu/ontologies/ELTER_CL/submissions/1/metrics',
  'properties': 0,
  'individuals': 579,
  'classes': 2,
  'maxChildCount': 2,
  'classesWithMoreThan25Children': 0,
  'classesWithOneChild': 0,
  'maxDepth': 0,
  'created': '2023-11-16T15:11:43+00:00',

  'submission': [
    'https://data.ecoportal.lifewatch.eu/ontologies/ELTER_CL/submissions/1'
  ],
  'classesWithNoDefinition': 2,
  'averageChildCount': 2,
  'numberOfAxioms': null,
  'entities': null,
  '@id': 'https://data.ecoportal.lifewatch.eu/ontologies/ELTER_CL/submissions/1/metrics',
  '@type': 'http://data.bioontology.org/metadata/Metrics',
  'links': {
    'ontology': 'https://data.ecoportal.lifewatch.eu/ontologies/ELTER_CL',
    'submission': 'https://data.ecoportal.lifewatch.eu/ontologies/ELTER_CL/submissions/1',
    '@context': {
      'ontology': 'http://data.bioontology.org/metadata/Ontology',
      'submission': 'http://data.bioontology.org/metadata/OntologySubmission'
    }
  },
  '@context': {
    '@vocab': 'http://data.bioontology.org/metadata/',
    'created': 'http://data.bioontology.org/metadata/created',
    'classes': 'http://data.bioontology.org/metadata/classes',
    'individuals': 'http://data.bioontology.org/metadata/individuals',
    'properties': 'http://data.bioontology.org/metadata/properties',
```

```
'maxDepth': 'http://data.bioontology.org/metadata/maxDepth',
'maxChildCount': 'http://data.bioontology.org/metadata/maxChildCount',
'averageChildCount': 'http://data.bioontology.org/metadata/averageChildCount',
'classesWithOneChild': 'http://data.bioontology.org/metadata/classesWithOneChild',
'classesWithMoreThan25Children': 'http://data.bioontology.org/metadata/classesWithMoreThan25Children',
'classesWithNoDefinition': 'http://data.bioontology.org/metadata/classesWithNoDefinition',
'numberOfAxioms': 'http://omv.ontoware.org/2005/05/ontology#numberOfAxioms',
'entities': 'http://rdfs.org/ns/void#entities',
'@language': 'en'
}
```

If a given semantic artefact has more than 200 classes identified for a metric, then that metric will provide an error message within the class list. Two messages can appear:

- **limitpassed:** it means that the 200-class limit was reached. When this error appears, it is followed by the total number of classes in the semantic artefact that matches this metric;
- **alltriggered:** it means that every class in the semantic artefact matched this metric.

■ Submissions

Submissions ⓘ {}

Version	Modified	Submitted	Actions
2.5 		September 13, 2023	Download
2.0 		May 18, 2020	Download

Figure 16: Submission table

In the submission area, all submissions made for that semantic artefact are available (Figure 16). The table includes four columns:

- **Version:** shows the different versions loaded into EcoPortal with the status next to it. The labels assigned automatically by the portal are: Parsed (the resource has been parsed), Indexed (the resource has been indexed), Metrics (the metrics for the resource have been calculated) and Annotator (the resource is ready to annotate

some text via [API](#) or the [Annotator tool](#)). EcoPortal indexes the latest submission. The previous submissions will appear with the "Archived" label, which means that submission will no longer be published;

- *Modified*: shows the latest modification date (mm/dd/yyyy) of the semantic artefact (manually set by the creator);
- *Submitted*: shows the original release date (mm/dd/yy) of the semantic artefact (manually set by the creator);
- *Actions*: allows the download of the semantic artefact in different formats SKOS, OBO, UMLS, OWL, CSV, RDF/XML, depending on the representation language of the resource. Furthermore, when more submissions occur for the same resource, the system performs an automatic download report (Diff) of the differences between the submitted versions.

■ Views

This area displays the semantic artefact's views and allows the creation of new ones. A view is a manageable portion of a larger semantic artefact that is tailored for specific applications and users.

When creating a view, the system:

- allows the precise extraction of subsets from a larger resource;
- allows to personalise and use portion of a resource;
- facilitates interoperability between subsets of large semantic artefacts and specific applications.

Each displayed view is linked for an easy consultation.

2.3.2 Menu

The menu area enables users to explore classes/concepts, properties, instances/schemes, collections (if available), notes, mappings, and available widgets through different tabs, according to the semantic artefact's formality level (Figure 17 Ontology view; Figure 18 Thesaurus view).

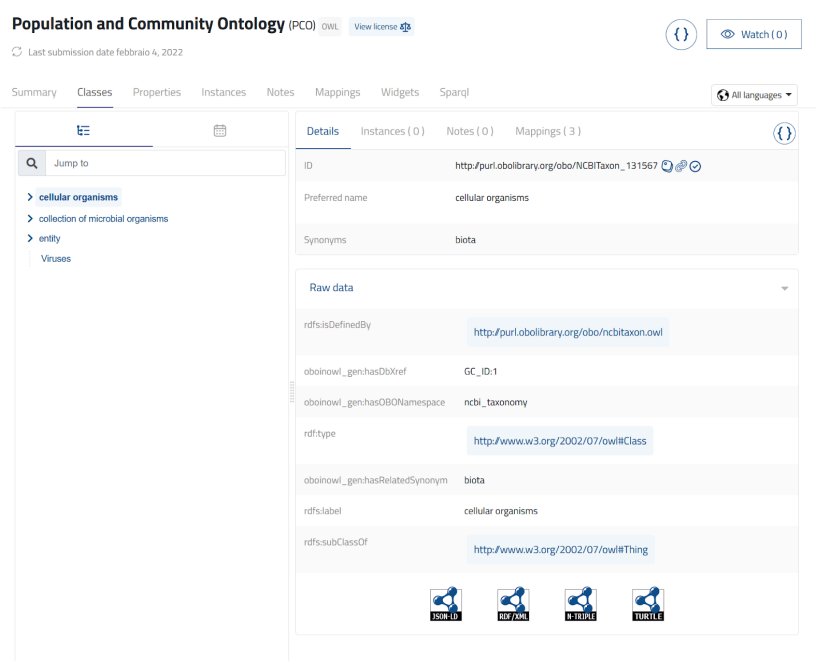
■ Classes/Concepts

On the summary page of the semantic artefact, by clicking on the tab "Classes/Concepts", it is shown:

- on the left side, classes/concepts can be displayed using a "Hierarchical

view", in which classes/concepts are organised in a tree-like structure, and a "Date view", which displays classes/concepts based on their creation and modification dates (if these were defined). In the case of a thesaurus (Figure 18) the "Collection view" will also be available. In addition, close to the search bar, there is a filter to select different schemes or collections eventually available within the thesaurus;

- on the right side, an area containing Details, Visualization, Notes and Mappings related to the selected class/concept is displayed:
 - **Details:** tab showing properties and relationships associated with the selected class/concept (Figure 17-18). Properties like ID, Preferred Name, Definition, and Type can be shown together with other properties, depending on the resource's representation model (SKOS; OWL; UMLS; OBO) and on the properties included in the selected semantic artefact.



Population and Community Ontology (PCO) OWL [View license](#)

Last submission date febbraio 4, 2022

Summary Classes Properties Instances Notes Mappings Widgets Sparql All languages

Details Instances (0) Notes (0) Mappings (3)

Jump to

- cellular organisms
- collection of microbial organisms
- entity
- Viruses

Details

ID	http://purl.obolibrary.org/obo/NCBITaxon_131567
Preferred name	cellular organisms
Synonyms	biota

Raw data

rdfs:isDefinedBy	http://purl.obolibrary.org/obo/ncbitaxon.owl
obo:owl_gen:hasDbXref	GC_ID:1
obo:owl_gen:hasOBONamespace	ncbi_taxonomy
rdf:type	http://www.w3.org/2002/07/owl#Class
obo:owl_gen:hasRelatedSynonym	biota
rdfs:label	cellular organisms
rdfs:subClassOf	http://www.w3.org/2002/07/owl#Thing

OWL RDFS RDF/XML R-TURTLE TURTLE

Figure 17: View classes of a specific ontology - details

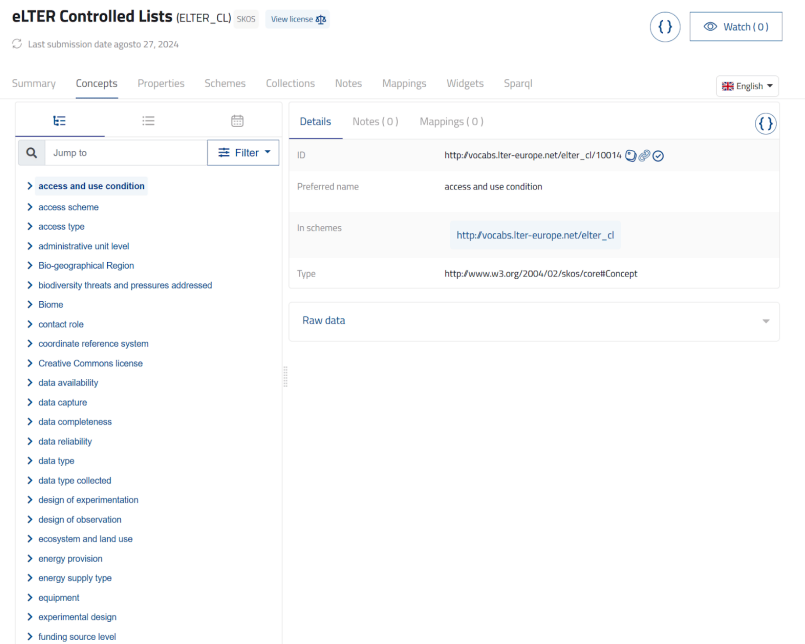


Figure 18: View classes of a specific thesaurus - details

- **Notes:** it displays all notes issued by EcoPortal users in relation to the selected class/concept. To issue generic comments or proposals regarding new classes, new relationships, or change properties, users must be logged in.
- **Class/Concept Mappings:** tab showing all the mappings associated with the selected class/concept (Figure 19). For each mapping users can see:
 - Mapping to: the name of the selected class/concept mapped to the class/concept of another resource;
 - Semantic resource: the semantic artefact of the mapped class/concept;
 - Type: the mappings can be internal to the portal or external to the portal;

Logged in users can click on the "Create new mapping" button to manually insert mappings by selecting the mapping type (i.e. Internal, InterPortal or External) and filling in the following fields: target class, details, mapping description, contact info, source name, comment, mapping relation type.

For more information about the mapping "Source" click [here](#).

eLTER Controlled Lists (ELTER_CL) SKOS [View license](#)

Last submission date August 27, 2024

Summary Concepts Properties Schemes Collections Notes Mappings Widgets Sparql English

Jump to Filter

- habitat classification
- INSPIRE data theme
- landform
 - Aeolian
 - Coastal and oceanic
 - Continental plain
 - Depression
 - Fluvial and limnic
 - Canyon
 - Delta
 - Lake
 - River
 - Mountainous
 - not applicable
 - location type
 - LTER site classification
 - measurement scale
 - metadata standards
 - method
 - MONERIS keyword
 - MONERIS variable

Details Visualization Notes (0) Mappings (8)

Create New Mapping

Mapping to	Semantic resource	Type
lake >	ENVTHES	LODM
lake >	SNOWTERM	LODM
Lake >	MRPTCODELIST	LODM
lake >	INBIO	LODM
lake >	BIODIVTHES	LODM
lake >	ENVTHES	LODM
lake >	EUROVOC	LODM
lake >	ENVTHES	LODM

Figure 19: View classed of a specific semantic artefact - mappings

■ Properties

In the summary page of the chosen semantic artefact, by clicking on the Properties tab all the properties used within the selected semantic artefact are displayed (Figure 20). Similar to the Classes/Concepts tab, the tree-like structure is shown on the left, while the property details are shown on the right.

Population and Community Ontology (PCO) OWL [View license](#)

Last submission date February 4, 2022

Summary Classes Properties Instances Notes Mappings Widgets Sparql English

Search a property in PCO

- 2D boundary of
- alternative term
- antisymmetric property
- bearer of
 - BFO_0000179
 - BFO_0000180
- causally related to
 - concretizes
 - contains process
 - contributor
 - coverage
 - created_by
 - creation_date
 - creator
 - creator note
 - database_cross_reference
 - date

Details

ID http://purl.obolibrary.org/obo/RO_0002000

Type <http://www.w3.org/2002/07/owl#ObjectProperty>

Preferred name 2D boundary of

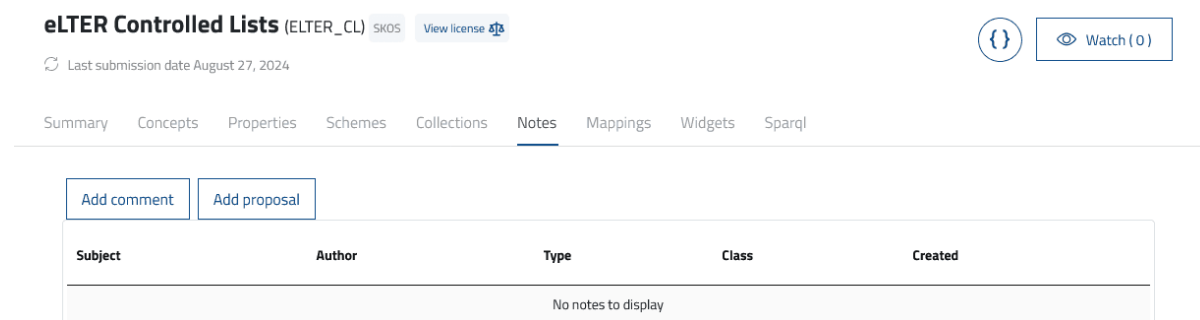
Definitions a relation between a 2D immaterial entity (the boundary) and a material entity, in which the boundary delimits the material entity

Raw data

Figure 20: View properties of a specific semantic artefact

■ Notes

In the Notes tab users can see all the comments or proposals issued by other registered EcoPortal users (Figure 21).



The screenshot shows the 'Notes' tab for the 'eLTER Controlled Lists' semantic artefact. At the top, there is a header with the artefact name, a 'SKOS' label, a 'View license' button, and a 'Watch (0)' button. Below the header is a navigation bar with tabs: Summary, Concepts, Properties, Schemes, Collections, Notes (selected), Mappings, Widgets, and Sparql. Under the 'Notes' tab, there are two buttons: 'Add comment' and 'Add proposal'. Below these buttons is a table with the following columns: Subject, Author, Type, Class, and Created. The table is currently empty, displaying 'No notes to display'.

Figure 21: View notes of a specific semantic artefact

- **Add a comment:** logged in users can send a comment to the owner of the resource by specifying the subject and writing a free text comment.
- **Add a proposal:** logged in users can send a proposal to the owner of the resource to modify or create new relationships or classes/concepts.

■ Mappings

The Mappings tab displays the mappings computed by EcoPortal for the selected resource, grouped by semantic artefact (Figure 22). The page includes a table with two columns: "Semantic artefact" and "Mappings" and a bubble plot. The first column identifies the semantic artefact to which the selected resource is mapped, and the second column shows the number of mappings. By clicking on a semantic artefact all the mappings are shown (Figure 23).

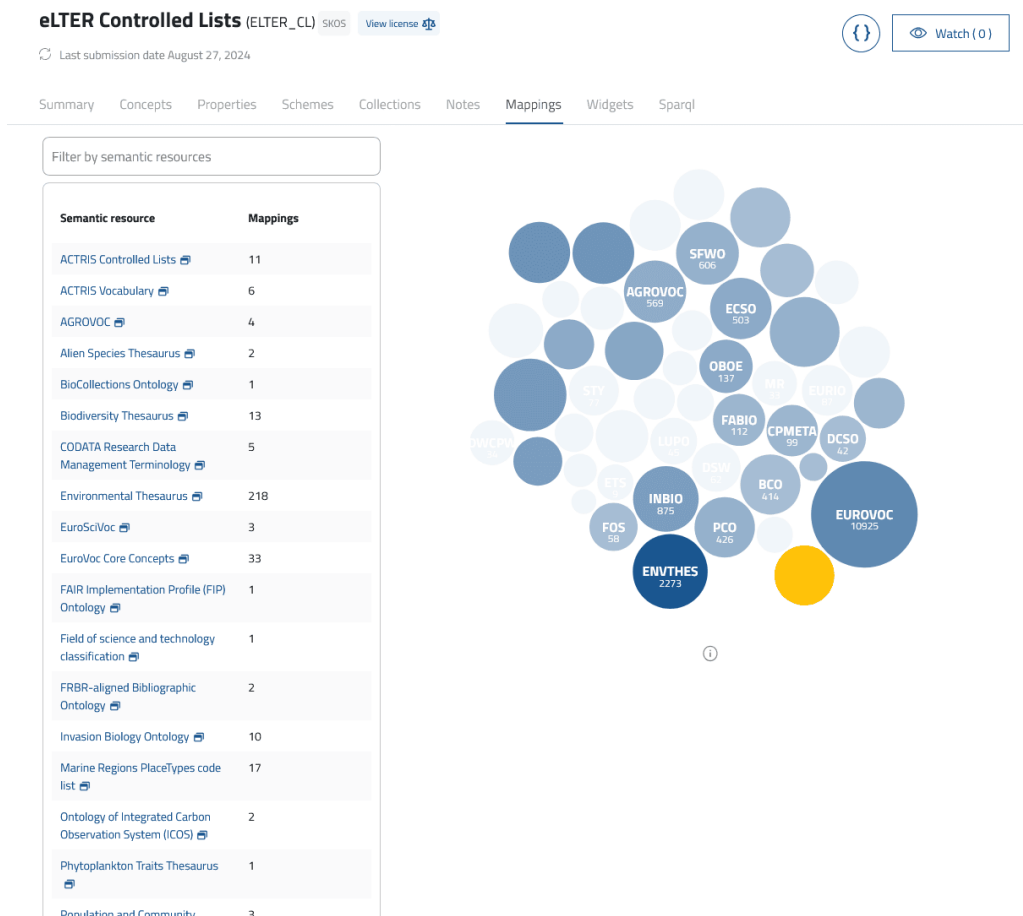


Figure 22: View mappings of a specific semantic artefact

ACTRIS Controlled Lists

ELTER_CL	ACTRIS_CL	Source
owner >	owner >	LOOM
user >	user >	LOOM
principalInvestigator >	principalInvestigator >	LOOM
processor >	processor >	LOOM
resource provider >	resourceProvider >	LOOM
publisher >	publisher >	LOOM
pointOfContact >	pointOfContact >	LOOM
originator >	originator >	LOOM

Figure 23: View mappings of a specific semantic artefact

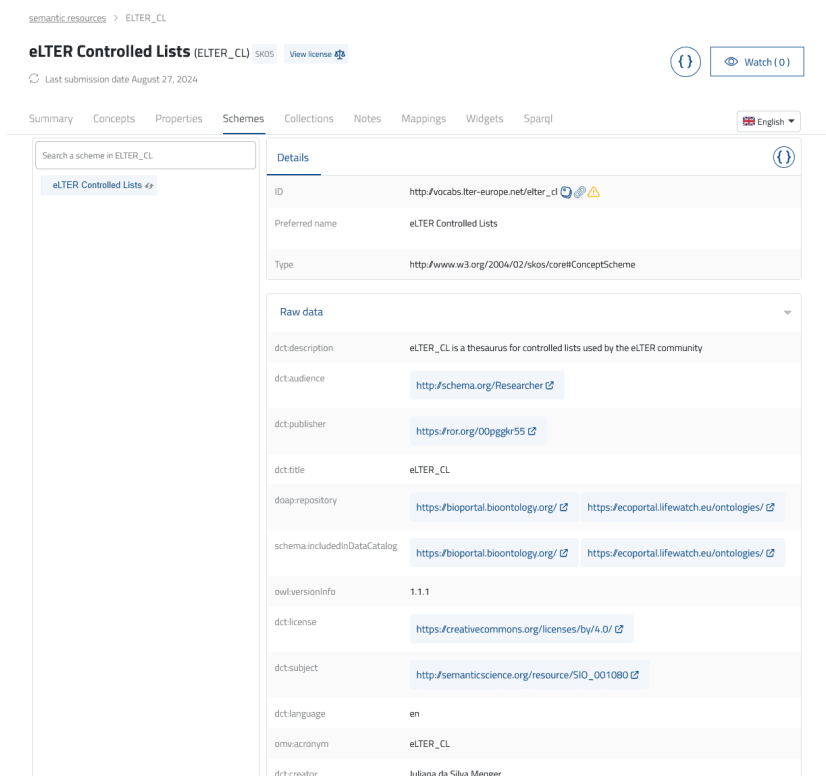
To access mappings, it is possible to click directly on the bubbles of the semantic artefact.

■ Instances/Schemes

In the Instances/Schemes tab, all the owl:NamedIndividuals and skos:conceptSchemes available in the selected artefact are displayed. By clicking on an instance/scheme more detailed information is found (Figure 24).

■ Collections

In the Collections tab, which is specific to thesauri, all the skos:collections available in the selected resource can be browsed. By clicking on them detailed information is shown (Figure 25).



The screenshot displays the 'eLTER Controlled Lists' (ELTER_CL) SKOS concept scheme details. The interface includes a search bar, a 'Watch (0)' button, and a 'Last submission date August 27, 2024' note. The 'Schemes' tab is selected, showing a list of schemes on the left and a detailed view on the right. The detailed view includes a 'Details' section with fields for ID, Preferred name, and Type. Below this is a 'Raw data' section with a table of properties and values.

Property	Value
dct:description	eLTER_CL is a thesaurus for controlled lists used by the eLTER community
dct:audience	http://schema.org/Researcher
dct:publisher	https://ror.org/00pggr55
dct:title	eLTER_CL
doap:repository	https://bioportal.bioontology.org/ https://ecoportal.lifewatch.eu/ontologies/
schema:includedInDataCatalog	https://bioportal.bioontology.org/ https://ecoportal.lifewatch.eu/ontologies/
owl:versionInfo	1.1.1
dct:license	https://creativecommons.org/licenses/by/4.0/
dct:subject	http://semanticscience.org/resource/SIO_001080
dct:language	en
omv:acronym	eLTER_CL
dct:creator	Juliana da Silva Menger

Figure 24: View schemes of a specific semantic artefact

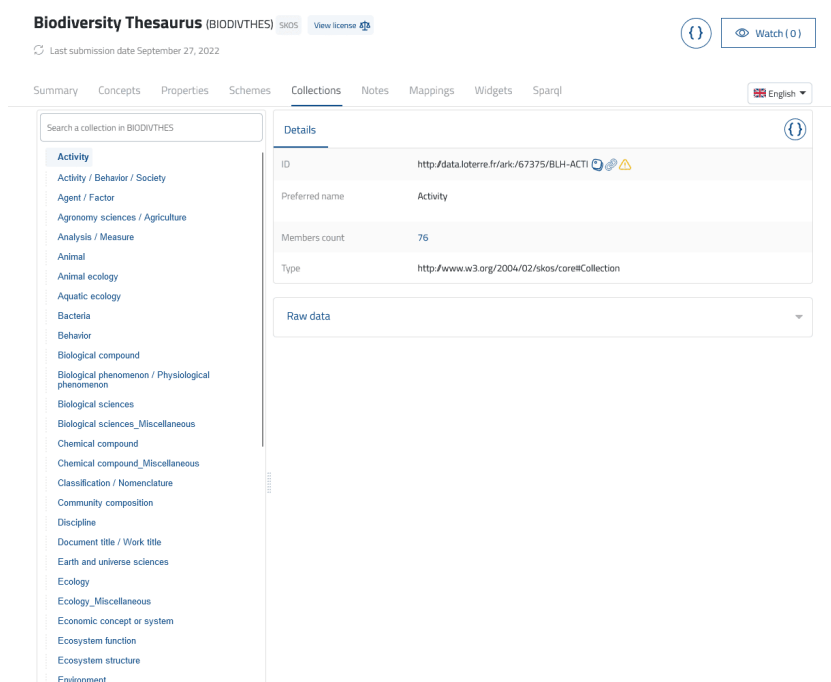


Figure 25: View collections of a specific semantic artefact

■ Widgets

Widgets tab contains the different types of widgets that can be incorporated in a web page (Figure 26). The page is divided in two sides, "Widget type" with the "Get code" buttons which lead to short instructions and to the download link of the code to be incorporated, on the right, their relative demos.

The types of widgets available in EcoPortal are:

- **Jump To:** type a class in the selected semantic artefact and jump to it in EcoPortal;
- **Autocomplete:** fill in a field in your form with a class from the selected semantic artefact;
- **Visualization:** display a visualisation for a given class/concept in the selected semantic artefact;
- **Tree Widget:** display a tree for classes in the selected semantic artefact with a search field.

■ SPARQL

Each semantic artefact has enabled the SPARQL endpoint, in which users can perform queries. The results can be viewed as a table or as an RDF/XML file structure. The results

can also be downloaded in RDF/XML.

Biodiversity Thesaurus (BIODIVTHES) SKOS [View license](#)

Last submission date September 27, 2022

Summary Concepts Properties Schemes Collections Notes Mappings **Widgets** Sparql

Add BIODIVTHES Web Widgets to your site

Jump To
Type a class name from BIODIVTHES and jump to it in EcoPortal

Widget See code {}

Jump To: [Go to EcoPortal](#)

Autocomplete
Fill your form fields with classes from BIODIVTHES

Widget See code {}

Example 1 (start typing the class name to get its full URI)

Example 2 (get the ID for a class)

Example 3 (get the preferred name for a class)

Visualization
Display a visualization for a given class in BIODIVTHES

Widget See code {}

Path to Root Fullscreen Screenshot Take a Tour Menu

Tree Widget
Display a class tree with a search field for BIODIVTHES

Widget See code {}

Search for class...

- applied ecology
- biological substance
- cultural heritage
- ecological parameter
- ecological process
- ecological system
- ecology
- environment
- environmental law
- environmental substance
- geographical zone
- international organization
- natural process
- natural product
- natural resource
- taxonomy

Figure 26: View widgets of a specific semantic artefact

Traits Thesaurus (TRAITS_THES) SKOS [View license](#)

Last submission date November 28, 2024

Summary Concepts Properties Schemes Collections Notes Mappings **Widgets** 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 * WHERE {
4   ?sub ?pred ?obj .
5 } LIMIT 10

```

Table Response

Figure 27: View of the SPARQL endpoint of a specific semantic artefact

■ Check Resolvability application

For each URIs available in the user interface, EcoPortal provides three options (Figure 28.1):

- **"Copy original URI";**
- **"Copy EcoPortal URI".** For each URIs, EcoPortal generates automatically an analogue 'EcoPortal URI' that does support HTTP content resolution and negotiation in 4 formats (JSON, XML, Turtle, N-Triple) and also HTML, bringing the users directly to the corresponding Web page in EcoPortal. These new URIs do not replace the original URI, and are not made to be used externally except to redirect the original URIs (when they are not resolvable and/or negotiable). On the semantic artefact summary page, in the Identifiers box, the EcoPortal URI (chain symbol), when clicked, will display the instructions (for Apache and Nginx Web servers) to use for the domain name redirections, if needed;
- **URI resolvability check.** This last option checks if the original URI is resolvable and negotiable (Figure 28.2, i.e., support HTTP content negotiation and return alternative formats). Users can "click to see details" and be moved to EcoPortal new Check Resolvability application (Figure 28.2).

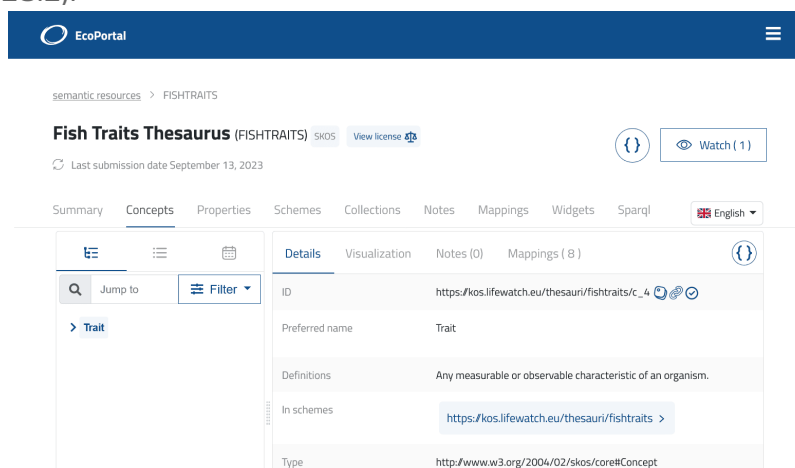


Figure 28.1: URIs inks and resolvability check.

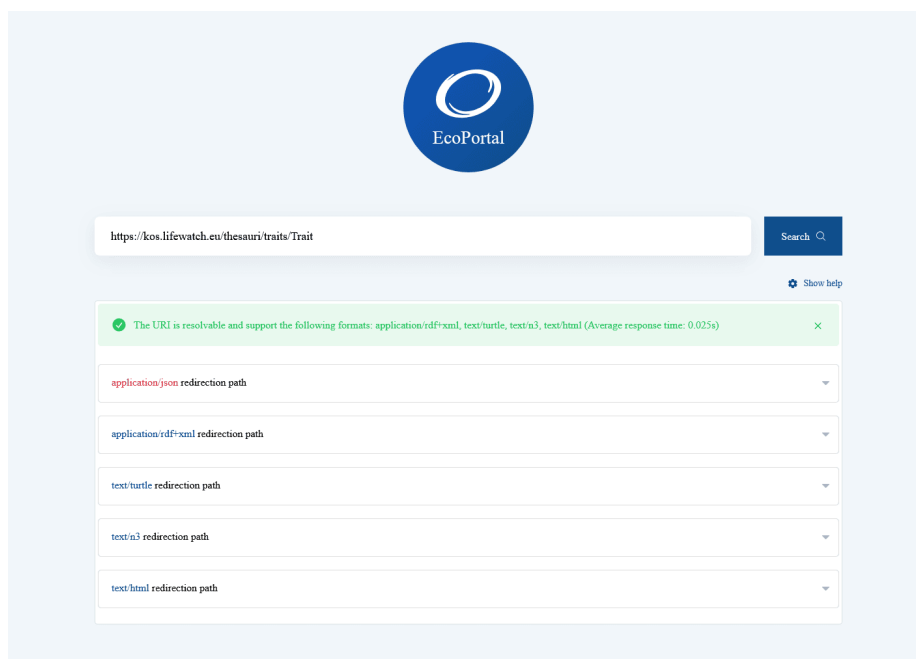


Figure 28.2: View of the Check Resolvability application.

3. EcoPortal Metadata Schema

The EcoPortal Metadata Schema is built upon the Metadata Vocabulary for Ontology Description and Publication ([MOD 2.0](#)). This comprehensive schema integrates various vocabularies, such as Dublin Core, OMV, DCAT, and VoID, and was introduced by [Dutta et al. 2017](#). Its primary purpose is to empower ontology developers to annotate and describe their ontologies, while also enabling ontology libraries to provide semantic descriptions of ontologies in the form of linked data.

EcoPortal with other members of the OntoPortal Alliance has embraced a set of metadata elements, comprising classes and properties organised into distinct categories. Each metadata element is linked to a FAIR principle, adding a layer of semantic richness to the description. The assessment of metadata elements is facilitated through the O'FAIRe tool, ensuring a systematic evaluation of the ontology's Findability, Accessibility, Interoperability, and Reusability aspects.

The table below lists all the metadata fields used in EcoPortal, and the FAIR sub-principles to which they are associated for the computation of the FAIR score.

Note: The column *Cardinality* indicates the quantity constraints for the metadata field. The possible values are:

- 0-n = optional and repeatable;
- 0-1 = optional, but not repeatable;
- 1-n = mandatory and repeatable;
- 1 = mandatory, but not repeatable.

Metadata category	Metadata element set	Definition	Cardinality	Associate d FAIR principle	MOD classes and properties	Example and notes
General	Acronym	The semantic artefact acronym.	1	F2	omv:acronym	
	Name	The semantic artefact name.	1		omv:name	
	Representation Language	Controlled list with four format type: SKOS, OBO, OWL, UMLS.	1	I1, I3	omv:hasOntologyLanguage	
	Categories	Categories to which the semantic artefact belongs.	0-n		omv:hasDomain	

	Groups	Group to which the semantic artefact belongs. Groups associate semantic artefact from the same project or organisation, for better identification of the provenance.	0-n	R1.3	bioportal:group	
	Administrators	Controlled list with registered user on EcoPortal.	1-n		bioportal:administeredBy	
	Version	The version number of the semantic artefact.	1	I3, R1.2	omv:version	Semantic versioning is highly recommended. To assign a version number follow the guidance below.

						Given a version number MAJOR.MINOR.PATCH, increment the: MAJOR version when you make incompatible API changes. MINOR version when you add functionality in a backward compatible manner. PATCH version when you make backward compatible bug fixes. Additional labels for
--	--	--	--	--	--	--

						pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.
	Status	Semantic artefact release status: alpha , the semantic artefact is actively in development after the previous release and being tested internally; beta , the semantic artefact is feature-complete and being tested	1	A2	omv:statuses	

		internally; production , the semantic artefact has passed all stages of verification and test; retired , the semantic artefact is no longer supported or is obsolete, and it will not be implemented any more.				
	Deprecat ed	To specify if the semantic artefact IRI is deprecate d.	0-1	A2	owl:depre cated	
	URI identifier	The URI of the semantic artefact which is	1	F1	omv:URI	

		described by this metadata				
	Version IRI	The property that identifies the version IRI of a semantic artefact.	1	F1	owl:versionIRI	
	Formality level	Level of formality of the semantic artefact.	1	I1, I3	omv:hasFormalityLevel	
	Ontology Syntax	The knowledge representation language for the semantic artefact language.	0-1	I1, I3	omv:hasOntologySyntax	
	Natural language	The language of the content of the semantic artefact.	0-n	F2	omv:naturalLanguage	
	Generic Type	The nature of the	0-1	F2	omv:isOfType	

		content of the semantic artefact.				
	Other identifier	An unambiguous reference to the semantic artefact. Use the semantic artefact URI if not provided in the semantic artefact metadata.	0-n	F1, A1	dct:identifier	
Description	Abstract	A summary of the semantic artefact.	0-1	F2	dct:abstract	
	Description	Description of the semantic artefact.	1	F2	dct:description	The best practice is to provide a single sentence. For a longer summary, use

						Abstract.
	Notes	Additional information about the semantic artefact that is not included somewhere else (e.g. information that you do not want to include in the documentation).	0-n	F2	omv:notes	
	Homepage	The URL of the homepage for the semantic artefact.	0-1	F2	foaf:homepage	
	Documentation	URL for further documentation.	0-1		omv:documentation	A journal article, white paper, blog, or news article cannot be provided as the

						document ation of the ontology.
	Keywords	List of keywords related to the semantic artefact.	0-n		omv:keyw ords	The keywords must be taken from controlled vocabulari es and the URIs must be attached and separated by a semicolon.
	Hidden label	A lexical label for a resource that should be hidden when generating visual displays of the resource, but should still be accessible to free text	0-n		skos:hidde nLabel	

		search operations	.			
	Alternative name	An alternative title for the semantic artefact.	1-n	F2	dct:alternative	Any alternative name for the semantic artefact is known in the community.
	Bibliographic reference	The URL of the bibliographic reference of the semantic artefact.	1-n	F2	bioportal:publication	
Dates	Creation date	Date of the semantic artefact first release.	1		bioportal:release	
	Validity date	Date (often a range) of validity of the semantic artefact.	0-1	F2	dct:valid	

	Curation date	The date the semantic artefact was curated.	0-1	F2	pav:curatedOn	
	Submission date	Date of the submission/release in the portal.	0-1		omv:creationDate	
	Modification date	Date of the last modification made to the semantic artefact.	0-1	F2	omv:modificationDate	
Licensing	License	Underlying licence model.	0-1	I3, R1.1	omv:hasLicense	
	Use guidelines	A related resource which defines how the semantic artefact should be used.	0-1	R1.1	cc:useGuidelines	
	More permissions	A related resource which	0-1	R1.1	cc:morePermissions	

		describes additional permissions or alternative licences.				
	Rights holder	The party holding the legal copyright to the semantic artefact.	0-n	R1.1	schema:copyrightHolder	E.g. LifeWatch Italy
Persons and organisations	Contact	A person or organisation owning or managing rights over the semantic artefact.	1-n	R1.1	dc:agent	
	Creator(s)	Main responsible for the creation of the semantic artefact.	0-n	R1.2	omv:hasCreator	A creator can not be an organisation.
	Contributors	Contributors to the creation of the	0-n	R1.2	omv:hasContributor	

		semantic artefact.				
	Curator	People who curated the semantic artefact.	0-n	R1.2	pav:curate dBy	
	Publisher	An entity responsible for making the semantic artefact available.	1-n	F2	dct:publis her	
	Funded or sponsored by	The organisation funding the semantic artefact development.	0-n	R1.2	foaf:funde dBy	
	Endorsed by	The parties that have expressed support or approval to this semantic artefact.	0-n	R1.3	omv:endor sedBy	
	Translator	Organisati	0-n	R1.2	schema:tr	

		on or person who adapted the semantic artefact to different languages, regional differences and technical requirements.			anslator	
Links	Location	The place from which the semantic artefact is retrieved: Metadata Only , choose this option if you want users to search and view only the semantic artefact metadata; Load from	1		bioportal:pullLocation	

		URL , new versions loaded daily; Upload Local File , choose a file from the local file system to upload the semantic artefact.				
	Is format of	URL to the original document that describes this semantic artefact in a not ontological format.	0-1	I1, I3	dct:isForm atOf	Indicate (if it exists) the URL where a semantic artefact is available in another format (PDF, CSV, XML, HTML). The document has to be produced before the publicatio n of the semantic artefact.

	Has format	URL to a document that describes this semantic artefact in a not ontological format generated from this semantic artefact.	0-n	I1, I3	dct:hasFormatOf	Indicate (if it exists) the URL where a semantic artefact is available in another format (PDF, CSV, XML, HTML). The document has to be produced after the publication of the semantic artefact.
	Download URL	An RDF dump, partial or complete, of a void:Dataset.	0-1	F2	void:dataDump	
	CSV dump	A CSV dump, partial or complete, of a void:Dataset.	0-1		bioportal:csvDump	

	URI lookup Endpoint	A protocol endpoint for simple URI lookup for the semantic artefact.	0-1	F2	void:uriLookupEndpoint	
	Free-text search endpoint			F2	void:openSearchDescription	
	SPARQL endpoint	Relates an instance of sd:Service to a SPARQL endpoint that implements the SPARQL Protocol service [SPROT] for the service. The object of the sd:endpoint property is an IRI.	0-n	A1	sd:endpoint	
	Source	A related resource from which the described	0-n	R1.2	dct:source	E.g. Direct link or citation to the original

		semantic artefact is derived.				research papers, databases , or existing semantic artefacts that were consulted or utilised during its development.
	Indexed or Included in catalog or repository	A data catalogue which contains this semantic artefact.	0-n	F4	schema:includedInDataCatalog	
	Indexed or Included in catalog or repository	A data catalogue which contains this semantic artefact.	0-n	F4	schema:includedInDataCatalog	
Media	The URL of a media associated to the ontology	A media object that encodes this semantic artefact. This property is	0-n	F2	schema:associatedMedia	

		a synonym for encoding.				
	Depiction	The URL of an image representing the semantic artefact.	0-n	F2	foaf:depiction	
	Logo	The URL of the semantic artefact logo.	0-1	F2	foaf:logo	
Community	Bug database	Link to the bug tracker of the semantic artefact (i.e. GitHub issues).	0-1	F2	doap:bugDatabase	
	Audience	Description of the target user base of the semantic artefact.	0-1	F2	dct:audience	
	Repository	Link to the source code repository.	0-1	F2	doap:repository	

	Mailing list	Mailing list home page or email address.	0-1	F2	doap:mailingList	
	To do list	Describes future tasks planned by a semantic artefact curator.	0-n	F2	voaf:ToDoList	
	Award	An award won by this semantic artefact.	0-n	F2	schema:award	
Usage	Known usage	The applications where the semantic artefact is being used.	0-n	F2	omv:knownUsage	Consider also declaring the projects, within EcoPortal, that are using the semantic artefact. E.g. Used to annotate traits in metadata.
	Design for task	The purpose	0-n	R1.2	omv:designedForOnt	

		for which the semantic artefact was originally designed.			ologyTask	
	Subject	A topic of the semantic artefact.	0-n	F2, I3	dc:subject	
	Coverage	The spatial or temporal topic of the semantic artefact, spatial applicability of the resource, or jurisdiction under which the semantic artefact is relevant.	0-1	F2	dct:coverage	E.g. Terminologies about Italian endemic species.
	Example of use	A reference to a resource that provides	0-n	F2	vann:example	

		an example of how this semantic artefact can be used.				
Relations	This ontology is a view of	A portion of a larger semantic artefacts.	0-1			
	Prior version	An URI to the prior version of the semantic artefact.	0-1	I3, R1.2	omv:hasPriorVersion	
	Has part (has views)	A related semantic artefacts that is included either physically or logically in the described semantic artefact.	0-n	F2, I3	dct:hasPart	
	Similar to	Vocabularies that are similar in scope and objectives,	0-n	I2, I3	door:similarTo	

		independently of the fact that they otherwise refer to each other.				
	Generally related to	A semantic artefact that uses or extends some class or property of the described semantic artefact.	0-n	I2	door:ontologyRelatedTo	
	Comes from the same domain	Semantic artefact that come from the same domain.	0-n	F2, I3	door:comesFromTheSameDomain	
	Has equivalences with	Semantic artefact that have an alignment which covers a substantial part of the described	0-n	I2, I3	door:isAlignedTo	

		semantic artefact.				
	Backward compatibl e	URI of a semantic artefact that has its prior version compatible with the described semantic artefact.	0-n	F2, I3	omv:isBac kwardCom patibleWit h	Compatibl e Version: when two versions of an SA are compatibl e, it means that changes made in the newer version do not disrupt the functionali ty or interopera bility with systems that use the previous version. For example, adding new concepts or properties

						to the ontology without removing or altering existing ones may result in compatible versions.
	Incompatible	URI of a semantic artefact that is a prior version of this semantic artefact, but not compatible .	0-n	F2, I3	omv:isIncompatible With	Incompatible Version: when two versions of a semantic artefact are incompatible, it means that changes made in the newer version introduce conflicts or inconsistencies with systems that rely on the

						previous version. Incompatible changes may include renaming or removing existing concepts, altering relationships between concepts, or changing the structure of the semantic artefact in a way that breaks existing implementations. Incompatible versions often require updating or
--	--	--	--	--	--	--

						modifying systems that use the ontology to ensure continued functionality.
	Disparate modelling with	URI of a semantic artefact that is considered to have a different model, because they represent corresponding entities in different ways e.g. instance in one case and a class in the other for the same concept.	0-n	F2, I3	door:hasDisparateModelling	A SKOS or an OWL corresponding artefact.
	Has disjunctions with	Indicates that the subject	0-n		voaf:hasDisjunctions With	

		vocabulary declares some disjunct classes with the object vocabulary .				
	Generalization of	Vocabulary that is generalised by some superclasses or superproperties by the described semantic artefact.	0-n	I2, I3	voaf:generalizes	
	Specialization of	If the semantic artefact is a latter version that is semantically equivalent to another semantic artefact.	0-n	I2	door:explanationEvolution	
	Imports	Indicates the imported	0-n	I2, I3	omv:userimports	

		semantic artefact into the semantic artefact described.				
	Used by	Semantic artefact that uses the described semantic artefact.	0-n	F2, I3	voaf:used By	A URI should be provided.
	Translated by	A semantic artefact that is a translation of the content of this semantic artefact.	0-n	F2, I3	schema:workTranslation	
	Translation of	The semantic artefact from which has been translated from.	0-n	I2, I3	schema:translationOfWork	
Content	Root of obsolete branch	Property used to specify the root of an	0-1		bioportal:obsoleteParent	

		obsolete branch in the semantic artefact.				
	URI Regex Pattern	A regular expression that matches the URIs of the semantic artefact entities.	0-1	F2	void:uriRegexPattern	
	Preferred namespace URI	The preferred namespace URI to use when using terms from this semantic artefact.	0-n	F2	vann:preferredNamespaceUri	
	Preferred namespace prefix	The preferred namespace prefix to use when using terms from this semantic artefact.	0-1	F2	vann:preferredNamespacePrefix	E.g. foaf, dct, lupu.

	Example of resource	An example identifier used by one item (or record) from a dataset	0-1	F2	idot:exampleIdentifier	E.g. https://kos.lifewatch.eu/thesauri/endemics/c_1
	Key classes	Representative classes/instances in the semantic artefact.	0-n	F2	omv:keyClasses	
	Metadata vocabulary used	Vocabularies that are used and/or referred to create .	0-n	I2	voaf:metadataVoc	E.g. http://purl.org/dc/terms .
Methodology	Knowledge representation paradigm	A representation formalism that is followed to describe knowledge in a semantic artefact.	0-1	R1.2	omv:conformsToKnowledgeRepresentationParadigm	E.g. First-order, Description Logic.
	Engineering	Information about	0-1	R1.2	omv:usedOntologyE	E.g. Waterfall,

	methodology	the method model used to create the semantic artefact.			engineering Methodology	Lifecycles, Agile (see also: https://www.mdpi.com/2079-9292/10/9/1060)
	Created With	Information about the tool used to create the semantic artefact.	0-n	R1.2	omv:used OntologyEngineering Tool	
	Accrual method	The method by which items are added to the semantic artefact.	0-n	R1.2	dct:accrual Method	E.g. Text mining, Bibliographic analysis.
	Accrual periodicity	The frequency with which items are added to the semantic artefact.	0-1	R1.2	dct:accrual Periodicity	E.g. every 6 months new terms could be added after an updated analysis.
	Accrual policy	The policy governing the	0-1	R1.2	dct:accrual Policy	E.g. Infrastructure

		addition of items to the semantic artefact.				policies that suggests when and how the semantic artefact should be updated.
	Competency question	A set of questions made to build a semantic artefact at the design time. They are used to clarify the scope, purpose, and requirements of the semantic artefact.	0-n	R1.2	mod:competencyQuestion	E.g. In a biodiversity ontology: What are the interactions between species within a given habitat?
	Was generated by	People who generated the semantic artefact.	0-n	R1.2	prov:wasGeneratedBy	
	Was	People	0-n	R1.2	prov:wasI	

	invalidated by	who invalidated the semantic artefact.			invalidated By	
Object description properties	Object preferred label property	Property used to specify objects preferred label.	0-1	I3, R1	bioportal:preferredLabelProperty	
	Object definition property	Property used to specify objects definition.	0-1	I3, R1	bioportal:definitionProperty	
	Object synonym property	Property used to specify objects synonyms.	0-1	I3, R1	bioportal:synonymProperty	
	Object author property	Property used to specify object author.	0-1	I3, R1	bioportal:authorProperty	
	Hierarchy property	A property that is used to specify the hierarchy.	0-1	I3, R1	bioportal:hierarchyProperty	
	Object obsolete	Property used to	0-1	I3, R1	bioportal:obsoleteProperty	

	property	specify obsolete objects.			operty	
	Object creation date property	Property used to specify the date of creation of a class or another object in the ontology.	0-1	I3, R1	bioportal:createdPro perty	
	Object modification date property	Property used to specify the date of modification of a class or another object in the ontology.	0-1	I3, R1	bioportal:modifiedP roperty	

EcoPortal administrators encourage the creators of semantic artefacts to compile the metadata fields by adding as much detail as possible to ensure the accurate communication of information to users and to be compliant to the FAIR principles when publishing a resource.

4. Semantic Artefact lifecycle

This section contains guidelines for managing semantic artefacts. EcoPortal tools enable users to handle the entire lifecycle of their semantic artefact, from the creation on

VocBench to the publication on EcoPortal.

- [Create and edit semantic artefacts](#)
- [Publish a semantic artefact from VocBench to EcoPortal](#)
- [Submit new semantic artefacts](#)
- [Update semantic artefacts](#)
- [Give feedback on semantic artefacts](#)

4.1 Create and edit semantic artefacts

[VocBench](#) is a service that enables the creation, editing and publishing of semantic artefacts.

To create a new project or edit an existing project in **VocBench** you have to log in through the LifeWatch Italy Single Sign On.

- a. **First time logging in.** Once your account is activated, you have to complete the log in inside VocBench (Figure 29).

Within the registration form, it is possible to request editing permission of a semantic artefact published in EcoPortal by adding the full resource name and acronym (e.g., [Fish Traits Thesaurus \(FISHTRAITS\)](#)) into the box "Existing semantic artefact?". By default, the last version on EcoPortal is loaded, otherwise users have to specify the version from which to start editing in VocBench. **NOTE:** Only the semantic artefact owner can request editing permission for that resource.

To create a new semantic artefact users must specify:

- *New semantic artefact:* the name of the project, e.g., Fish Traits Thesaurus. It is recommended to add the full name of the resource that will appear in EcoPortal;
- *BaseURI:* the URL necessary for the creation of a project. **NOTE:** if users do not have a BaseURI, the system will provide the following:
 - <https://kos.lifewatch.eu/ontologies/yourACRONYM> for ontologies;
 - <https://kos.lifewatch.eu/thesauri/yourACRONYM> for thesauri, glossaries or other loose sets of concepts.
- *Model:* the model of the new project which can be [OWL](#) or [SKOS](#). **NOTE:** To understand more about the SKOS model in EcoPortal it is recommended to read the following [section](#).

- b. **Registered users** can contact the Administrators at this email address semantics@lifewatch.eu, specifying: 1. name of the new semantic artefact, 2. Base URI (check the previous paragraph for more specific information), 3. Model (e.g. OWL, SKOS, RDFS, OntoLex, EDOAL), 4. Lexicalization (e.g. RDFS, SKOS, SKOS-XL, OntoLex), 5. specify which among history, validation, blacklisting, undo, trivial inference to have, 6. repository access (e.g. create a local project or a remote one), 7. configuration (e.g. native store, in memory, GraphDB). The request is then managed by the administrator that will create the new project and assign the role of [Project Manager](#) to the creator.

In the list of projects, there should be a new item for the project as shown in Figure 30, with information related to name, model and lexicalization of the resource. It is indicated whether history and validation have been activated in the project. Lastly, the repository location is specified (e.g. local or remote).

VocBench

About VocBench en

User registration:

E-mail: *

Password: * Confirm password: *

Given name: *

Family name: *

Office address:

Phone:

Affiliation:

Personal URL:

Is your semantic asset already in EcoPortal and you would like to edit its content? Please provide us its title as it appears in EcoPortal:

Would you like to create a new semantic asset? Please provide a name for it:

What is the baseURI of your semantic asset? If you do not have one yet, you may use http://www.example.org:

If this is a new asset you are creating, please indicate the model you would like to use: OWL or SKOS?:

User IRI:

Avatar URL:

Language proficiencies: **

* Mandatory field
** Optional but highly recommended field

Submit

Figure 29: Registration in VocBench and creation of a new semantic artefact.

Projects list

Name	Model	Lexicalization	History	Validation	Repository Location
TestThesaurusDeployerEcoPortal	SKOS	SKOS	×	×	local

Access Close

Figure 30: List of projects in VocBench.

All subsequent steps require moving to the "Data" tab to edit the semantic artefact. In the "Scheme" tab (Figure 31), click on the create button to create a new skos:ConceptScheme ([]*).

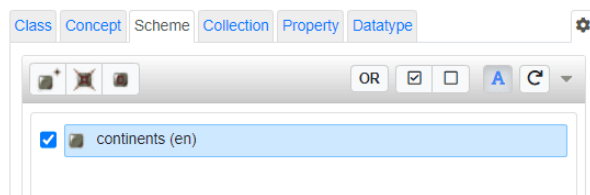


Figure 31: Scheme tab of the projects in VocBench.

In the scheme creation dialog (Figure 32), specify the label of the concept scheme e.g., "continents" in English and then click on the **Ok** button.

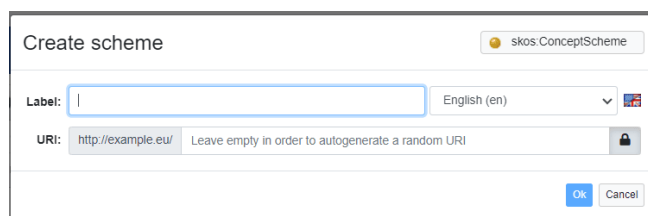


Figure 32: Creation of a concept scheme in VocBench.

Then, select the newly created scheme as in Figure 33.

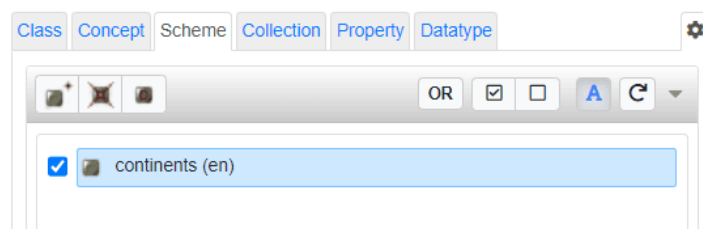


Figure 33: Selection of a newly created scheme in VocBench.

In the Concept tab (Figure 34), click on the create button to create a new skos:Concept ()*).



Figure 34: Concept tab of the project in VocBench.

In the concept creation dialog (Figure 35), specify the label of the concept e.g., "Africa" in English and then click on the Ok button. This label represents the skos:prefLabel of the concept.

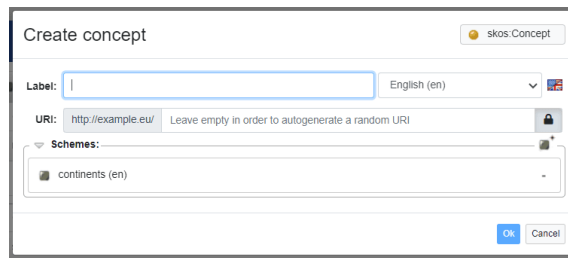


Figure 35: Creation of a new concept in VocBench.

Do the same to create other concepts:

- America
- Asia
- Europe
- Oceania
- Antarctica

The concept tree generated should contain a flat list of the five continents (Figure 36). It should be noted that the list of concepts generated is visualised in alphabetical order.

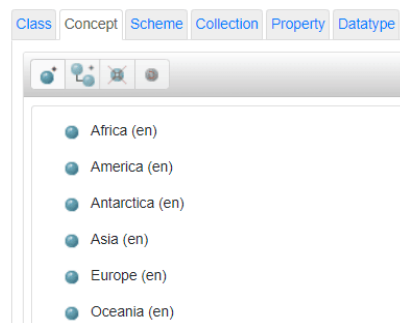


Figure 36: The concept tree in VocBench.

To add hierarchical depth to the structure, in the concept tab (Figure 37) select one continent and click on "Create narrow concept" button (()*).



Figure 37: Create narrow concept.

In the creation dialog (Figure 38), specify the label for the narrow concept e.g., "Italy" in English and then click on the **Ok** button.

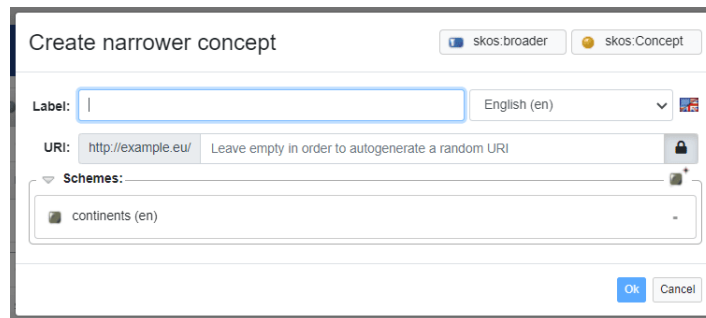


Figure 38: Wizard for the creation of narrower concept.

To understand in detail the different aspects of the user experience on VocBench, we recommend reading the following [User Manual](#).

Import an excel into VocBench

VocBench uses the platform **Sheet2RDF** to generate RDF triples from datasheets. Sheet2RDF can be accessed by clicking on the Tool tab of the VocBench menu and selecting the Sheet2RDF entry.

To use the Sheet2RDF tool correctly, it is essential to compile with the correct formatting whereby each datasheet column represents a SKOS/OWL property. An illustrative example of a valid file format is provided [here](#). For a comprehensive understanding of the user interface and the diverse functionalities of the tool, please consult this [resource](#). The following example provides details about the SKOS concepts creation when using Sheet2RDF.

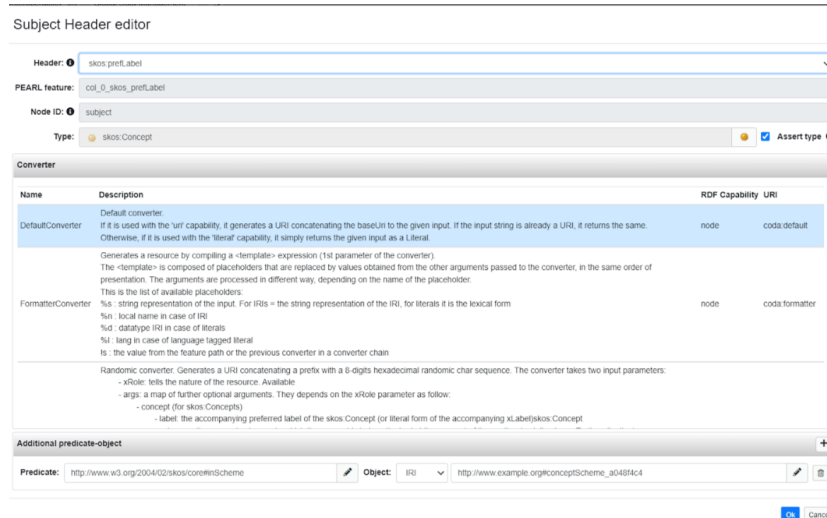
Example:

After uploading the Excel file using the "Browse" button, select "Subject mapping" in red (see Figure 39). Users are then required to:

1. Select the header that is going to be your Subject;
2. Select on the "Assert type" check box;
3. Click on the yellow circle and select "Concept";
4. In the "Converter" tab below, select "DefaultConverter";
5. In the "Additional predicate-object" tab, click on the "plus" button on the side;
6. Add the Predicate by selecting the SKOS property "inScheme";

7. Then select the Object by clicking on "Concept Scheme" and then on the desired scheme;
8. Then click "OK".

Note: the Concept Scheme has to be created before uploading the SKOS concepts.



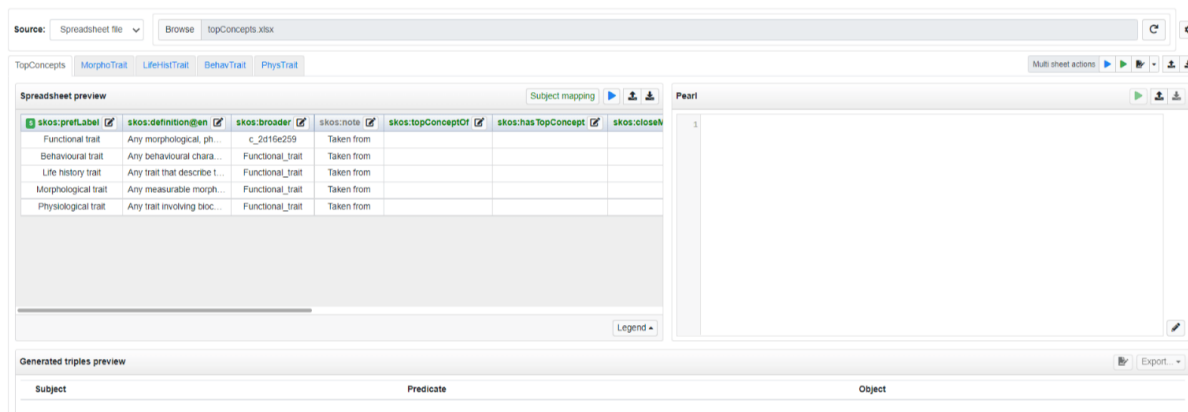
Name	Description	RDF Capability	URI
DefaultConverter	Default converter. If it is used with the 'uri' capability, it generates a URI concatenating the baseUri to the given input. If the input string is already a URI, it returns the same. Otherwise, if it is used with the 'literal' capability, it simply returns the given input as a Literal.	node	code:default
FormatterConverter	Generates a resource by compiling a <template> expression (1st parameter of the converter). The <template> is composed of placeholders that are replaced by values obtained from the other arguments passed to the converter, in the same order of presentation. The arguments are processed in different way, depending on the name of the placeholder. This is the list of available placeholders: %s - string representation of the input. For IRIs - the string representation of the IRI, for literals it is the lexical form %n - local name in case of IRI %t - datatype IRI in case of literals %l - lang in case of language tagged literal %v - the value from the feature path or the previous converter in a converter chain	node	code:formatter
RandomicConverter	Randomic converter. Generates a URI concatenating a prefix with a 8-digits hexadecimal random char sequence. The converter takes two input parameters: - xRole: tells the nature of the resource. Available - args: a map of further optional arguments. They depends on the xRole parameter as follow: - concept (for skos:Concept) - label: the accompanying preferred label of the skos:Concept (or literal form of the accompanying xLabel:skos:Concept		

Additional predicate-object

Predicate: Object: URI:

Figure 39: Subject header editor.

Upon executing these tasks, all headers should be highlighted in **green** (Figure 40).



Subject	Predicate	Object

Figure 40: Sheet2RDF tool.

If a header appears in **yellow** or **black**, by clicking on the pencil icon next to the header users have the possibility to either manually map the SKOS/OWL property or ignore the

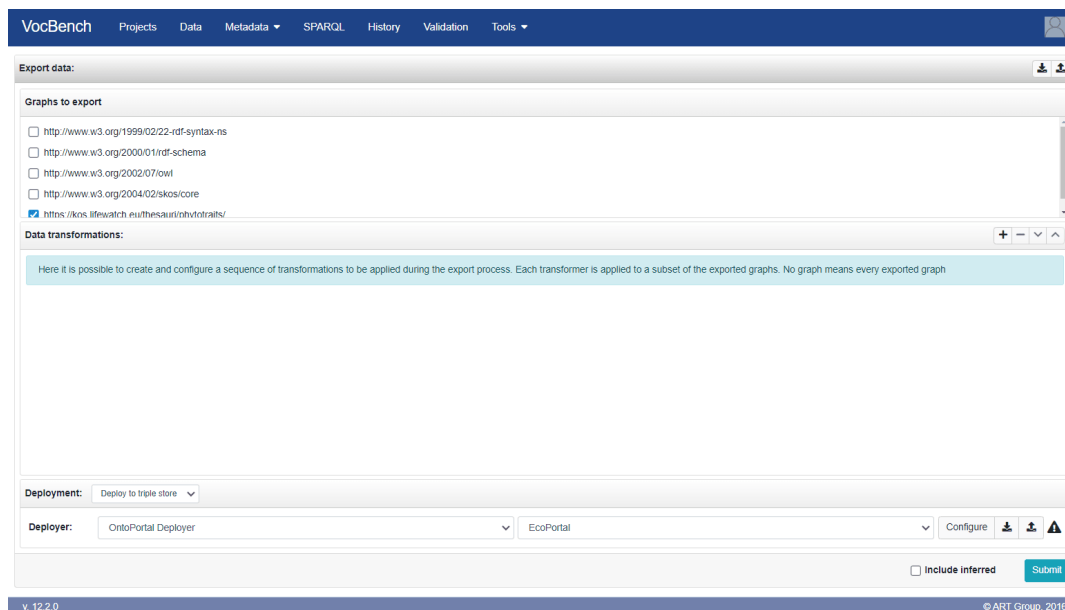
header. Following this, the blue "play" button and the subsequent green one should be clicked to generate the PURL code and triples. After the triples are generated, navigating to the "Generated triples preview" tab allows users to click on the "Add triples" button, presenting all the information in the "Data" section.

4.2 Export a semantic artefact from VocBench

To facilitate the publication of semantic artefacts in EcoPortal, a direct deployer from VocBench has been constructed. Users have to access the "Global Data Management" menu and click on "Export data".

The "Export data" panel consists of three sections as shown in Figure 41:

- *Graphs to export*: this section allows the selection of the graph to export. The default selection just contains the graph of the selected project, named with the base URI, which contains the data being edited.
- *Data transformations*: this section allows the configuration of a sequence of RDF Transformers, which can be used to manipulate the data being exported (e.g. selecting a specific ConceptScheme, add triples, replace triples, etc.)
- *Deployment*: this section allows the configuration of the data destination. Users can choose from a dropdown menu whether the data will be saved to file, exported to a triple store, or sent to a custom destination. The target of the EcoPortal Deployer is considered a triple store.



VocBench Projects Data Metadata SPARQL History Validation Tools

Export data:

Graphs to export

- ☐ http://www.w3.org/1999/02/22-rdf-syntax-ns
- ☐ http://www.w3.org/2000/01/rdf-schema
- ☐ http://www.w3.org/2002/07/owl
- ☐ http://www.w3.org/2004/02/skos/core
- ☒ https://kos.lifewatch.eu/thesauri/ontology/

Data transformations:

Here it is possible to create and configure a sequence of transformations to be applied during the export process. Each transformer is applied to a subset of the exported graphs. No graph means every exported graph

Deployment: Deploy to triple store

Deployer: OntoPortal Deployer EcoPortal

Configure

☐ Include inferred

Submit

v. 12.2.0 © ART Group, 2016

Figure 41: Global Data Management export.

After selecting "Deploy to a triple store" in the Deployment dropdown, the Deployer dropdown list will appear. Here users have to select "OntoPortal Deployer" and the item labelled EcoPortal to enable the specific characteristics of the deployer (e.g., additional configuration parameters) dedicated to EcoPortal.

The warning sign on the right of the widget associated with the EcoPortal Deployer indicates that it requires further configuration. A click on the "Configure" button will open the dialog shown in Figure 40 to edit the chosen configuration. Below is presented a summary of the EcoPortal configuration, which consists of several fields:

- *API Base URL*: the base URL of the EcoPortal REST API. If this parameter is omitted, the users can paste the following deployer defaults to the official base URL of EcoPortal: <http://ecoportal.lifewatch.eu:8080/>.
- *API key (mandatory)*: the API key that will be used to authorise the semantic artefact submission. User API key can be found in the [user account page](#).
- *Acronym (mandatory)*: the acronym that identifies the semantic artefact for which the submission is being made.
- *Description*: a textual description of the semantic artefact.
- *Version*: the version of the semantic artefact.
- *Format (mandatory)*: the format of the semantic artefact (either OWL or SKOS).
- *Status (mandatory)*: the status of the semantic artefact: **alpha**, the semantic artefact is actively in development after the previous release and being tested

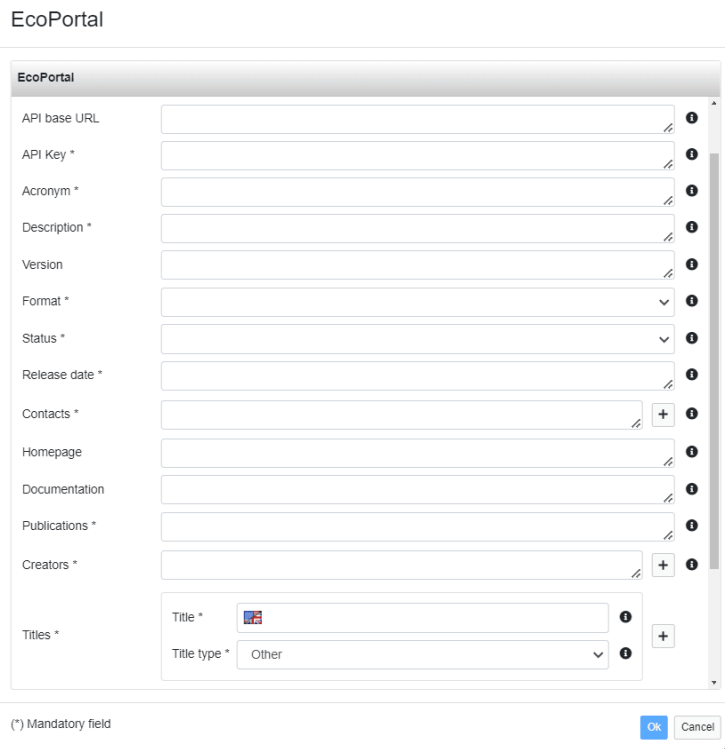
internally; **beta**, the semantic artefact is feature-complete and being tested internally; **production**, the semantic artefact has passed all stages of verification and test; **retired**, the semantic artefact is no longer supported or is obsolete, and it will not be implemented any more.

- *Release Date (mandatory)*: the release date of the semantic artefact, formatted as yyyy-mm-dd.
- *Contacts (mandatory)*: at least one contact for the semantic artefact is required. Additional contacts can be added by clicking on the plus button on the right. Each contact should be formatted as follows: *Name Surname (email)*. Example: Mario Bianchi (mario.bianchi@example.org).
- *Homepage*: the address of the main web page of the semantic artefact.
- *Documentation*: the address of a web page providing documentation for the semantic artefact.
- *Publications (mandatory)*: the address of a web page listing publications for the semantic artefact.
- *Creators (mandatory)*: people involved in the creation of the resource.
- *Titles (mandatory)*: the semantic artefact name or title.
- *Publisher (mandatory)*: the name of the entity that holds, archives, publishes, prints, distributes, releases, issues, or produces the semantic artefact.
- *Publication year (mandatory)*: the year when the resource was or will be made publicly available.
- *Resource type (mandatory)*: a description of the semantic artefact. The format is open, but the user can choose among the drop down list available (i.e. Authority file, Controlled Vocabulary, Glossary, Ontology, Thesaurus).
- *Resource type general (mandatory)*: the general type of resource.

The fields marked with an asterisk (*) are mandatory and all other fields are optional. Once the deployer has been configured, users can click on the "submit" button (Figure 42).

Note: We encourage creators, managers and editors of the semantic artefact in continuing the editing of their metadata directly from the EcoPortal edition page.

EcoPortal



(*) Mandatory field

OK Cancel

Figure 42: EcoPortal configuration wizard.

4.3 Submit new semantic artefacts

In order to publish a semantic artefact, you need to sign up for an account at the EcoPortal website:

- Login EcoPortal: [link](#)

Logged in users can navigate to the [Browse page](#) and click the [Submit a semantic artefact](#) button. The submission form requires to fill out some [metadata](#) about the submitted semantic artefact as shown in Figure 43:

- *Name*: name of the semantic artefact.
- *Acronym*: semantic artefact acronym (**NOTE**: It is not possible to change the semantic artefact acronym afterwards).
- *Administrators*: the owners of the resource. Only registered users can be added as administrators of the semantic artefact.
- *Viewing restriction*: choose if the submitted resource is public (all users, even unregistered users, can see it) or private.

- *Categories*: select the categories to which the resource belongs to improve the search tools.
- *Groups*: select the groups to which the resources belong. Groups associate ontologies from the same project or organisation, for better identification of the provenance.
- *View*: check "this semantic artefact is a view of" to create a new semantic artefact as a [view](#) of a semantic artefact selected from the drop-down field.
- *URI*: The URI of the semantic artefact which is described by this metadata.
- *Description*: description of the semantic artefact.
- *Representation language*: the model used to create the semantic artefact (e.g. SKOS, OWL, OBO).
- *Status*: semantic artefact release status: **alpha**, the semantic artefact is actively in development after the previous release and being tested internally; **beta**, the semantic artefact is feature-complete and being tested internally; **production**, the semantic artefact has passed all stages of verification and test; **retired**, the semantic artefact is no longer supported or is obsolete, and it will not be implemented any more.
- *Location*: the place from which the semantic artefact is retrieved: **Metadata Only**, choose this option if you want users to search and view only the semantic artefact metadata; **Load from URL**, new versions loaded daily; **Upload Local File**, choose a file from the local file system to upload the semantic artefact.
- *Date of the original creation*: date of the semantic artefact first release.
- *Contact*: details of a contact person (name and email).

Check the "Subscribe to email notifications for new notes" box to receive updates from users of the EcoPortal community.

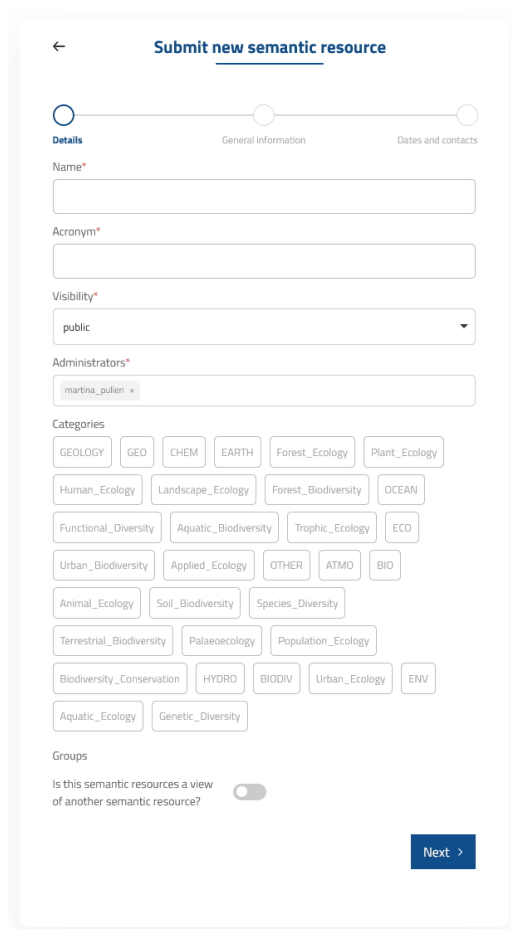


Figure 43: Wizard for the submission of a new semantic artefact.

Once the mandatory fields are completed, click on the "Finish" button in order to generate the summary page of the semantic artefact. Then click on the pencil button on the top right side of the summary page (Figure 45) to edit metadata and add more information about the semantic artefact (Figure 44).

To request the DOI, you can check the box "*Check this box if your asset has no PID and you want to request a DOI*", which can be found in the "Other identifier" metadata field.

LifeWatch ERIC is a member of [DataCite](https://datacite.org/) and it can provide a [Digital Object Identifier \(DOI\)](https://www.datacite.org/) to semantic artefacts that do not have one. DOIs allow semantic artefacts to be cited in scientific publications or elsewhere in a reliable and sustainable way. EcoPortal assigns a DOI only upon verification and validation of the semantic artefact. The DOI is connected to

the metadata of the resource as well as to its digital location (e.g. the URL) where all the details about the resource are available.

For more detailed information about metadata properties, refer to the [General Information](#) of the Metadata schema table.

Once the semantic artefact is submitted, users can immediately access the metadata, whereas the classes/properties indexing takes place asynchronously and it might be scheduled overnight. When the parsing of the semantic artefact is completed, an email is sent to the administrators of the semantic artefact. The whole submission should be completed by the next day; if not, please send an email to the support team (semantics@lifewatch.eu).

4.4 Update semantic artefacts

Logged in administrators of semantic artefacts can access the [summary_page](#) in the menu area to find the following options:

- Edit submission;
- Add submission.

Edit semantic resource

Filter properties to show

Start typing to select properties

General

Description

Dates

Licensing

Persons and organizations

Links

Media

Community

Usage

Relations

Content

Methodology

Object description properties

Edit the metadata of your semantic resource here. Some of these values are used by EcoPortal functionalities, including for FAIRness assessment. [See guidelines and recommendations for metadata here.](#)

Acronym*

TRAITS_THES

Name*

Traits Thesaurus

Representation language* ⁽ⁱ⁾

SKOS

SKOS vocabularies submitted to EcoPortal shall follow a few constraints (e.g., contain a minimum of one skos:ConceptScheme also typed as owl:Ontology) and top concept assertion. [Please refer to the documentation for more details.](#)

Categories

GEOLOGY GEO CHEM EARTH Forest_Ecology Plant_Ecology Human_Ecology Landscape_Ecology

Forest_Biodiversity OCEAN ☒ Functional_Diversity ☒ Aquatic_Biodiversity Trophic_Ecology ☒ ECO

Urban_Biodiversity Applied_Ecology OTHER ATMO ☒ BIO ☒ Animal_Ecology Soil_Biodiversity

Species_Diversity Terrestrial_Biodiversity Palaeoecology Population_Ecology Biodiversity_Conservation HYDRO

☒ BIODIV Urban_Ecology ENV ☒ Aquatic_Ecology Genetic_Diversity

Groups

Administrators*

[martina_pulieri](#) x

Version information ⁽ⁱ⁾

1.0.0

For more information on how to encode versioning information in a semantic resource, see [guidelines and recommendations](#).

Status* ⁽ⁱ⁾

beta

Deprecated ⁽ⁱ⁾ ☐

A semantic resource with status retired shall necessarily be also deprecated, but not the opposite.

URI* ⁽ⁱ⁾

<https://kos.lifewatch.eu/thesauri/traits/>

Version IRI ⁽ⁱ⁾

Formality level ⁽ⁱ⁾

Thesaurus

Syntax ⁽ⁱ⁾

RDF/XML

Natural language ⁽ⁱ⁾

English x

Generic type ⁽ⁱ⁾

Vocabulary

Other identifier ⁽ⁱ⁾

<https://doi.org/10.48373/sa6p-ta25>

+ Add another

x Cancel

✓ Save

Figure 44: Wizard for adding/editing metadata of the submission for semantic artefacts.

Edit submission

The editing of semantic artefacts metadata can be performed in two ways: edits of a specific version of the resource ("Pencil" button on the bottom left corner, Figure 45.2), edits of the latest submission ("Pencil" button on the top right corner, Figure 45.1). It is not possible to change the semantic artefact acronym because EcoPortal uses this field as a primary key and identifier. Once the form is edited, click on the "Save" button. If the "Location" settings are not changed, only the metadata changes will be saved.

Add submission

From this section, located at the top right corner (see Figure 45.1), the semantic artefact manager(s) can upload a new version of the existing semantic artefact and update the metadata using one of the following two options:

- If in the original submission was set up an automatic update of the resource from a source URL, EcoPortal examines the content of this URL nightly, and re-submits the content if it has changed. If the updated submission does not parse, the semantic artefact manager(s) will be notified via email;
- If in the original submission was not set up an automatic update of the resource, or if the semantic artefact manager(s) want to immediately update the semantic artefact, they can upload the new version directly by selecting a local file. The upload process starts and the semantic artefact will be parsed shortly thereafter. Semantic artefact manager(s) can at this point also change the settings and enable the automatic update of the resource. The semantic artefact parsing is completed within an hour maximum with the exception of large semantic artefacts, whereas semantic artefact indexing and automatic mapping occur overnight.

4.5 Give feedback on semantic artefacts

EcoPortal includes some community-oriented features in the [notes](#) such as: (i) comments that can be attached in a forum-like mode to a specific artefact or class/concept, in order to discuss different specifications of the semantic artefact (its design, use, or evolution) or allow users to propose changes, and (ii) proposals that can be submitted within the portal.

Administrators encourage the users to engage in giving feedback on semantic artefacts since each input is valuable in developing and publishing community-approved resources.

semantic resources > ALIENSPECIES

Alien Species Thesaurus (ALIENSPECIES) SKOS [View license](#)

Last submission date September 13, 2023

Summary Concepts Properties Schemes Collections Notes Mappings Widgets Sparql

General information

The Alien Species Thesaurus is an initiative to deal with the semantics about alien and invasive species. It has been developed and published by LifeWatch Italy, the Italian node of the e-science European infrastructure for biodiversity and ecosystem research (LifeWatch ERIC). It is developed on the basis of already existing glossaries created within the framework of international initiatives aimed at the study of alien and invasive species. The terms (or... See more...

Initial created on **May 18, 2020**. For additional information, contact **Ilaria Rosati** (ilaria.rosati@cnr.it) and **Angela Boggero** (angela.boggero@cnr.it).

Categories and subjects

Eco Biodiv Env

Share on socials [f](#) [x](#) [in](#)

[Download](#) [Home](#) [Bookmarks](#) [Refresh](#) [Share](#) [Print](#) [Help](#)

[Export all metadata](#) | [Export to DataCite](#)

Identifiers

URI
<https://kos.lifewatch.eu/thesauri/alienspecies/>

EcoPortal URI
<http://ecoportal.lifewatchdev.eu/ontologies/ALIENSP>

DOI
<https://doi.org/10.48373/xb7h-dw52>

Dates

Persons and organizations

Other links

Projects and usage information

Methodology and provenance

Community

Content

Metrics

2 Classes
165 Individuals
0 Properties

FAIR score

Beta

Total score : 266.0 (55.0%)

Findable
Accessible
Interoperable
Reusable

Obtained score
Not obtained score
Not yet supported

See details

Submissions

ID	Version	Modified	Submitted	Actions
2	2.5		September 13, 2023	Download Edit
1	2.0		May 18, 2020	Download Edit Delete

Views of ALIENSPECIES

No views available for ALIENSPECIES.

Configuration metadata

Figure 45: 1. Editing section 2. Submission section.

4.6 SKOS Support

Support for SKOS vocabularies in EcoPortal

EcoPortal serves as a web-based portal designed for the access and exchange of semantic artefacts. The platform is capable of receiving semantic artefact submissions in OWL and OBO formats, as well as SKOS vocabularies containing specific constructs. This

guide outlines the essential set of SKOS constructs required in a vocabulary for proper acceptance and handling within EcoPortal. It is important to emphasise that the described SKOS constructs are specifically processed for vocabularies identified as SKOS during their submission to EcoPortal.

Required SKOS constructs

- **skos:Concept.** Concepts are the fundamental elements of SKOS vocabularies and are asserted using the `skos:Concept` class, e.g.:
`<http://www.example.com/animals> rdf:type skos:Concept`

In SKOS vocabularies, EcoPortal only treats the SKOS concept assertions as concepts to be displayed. If the vocabulary contains other assertions about other types of concepts, EcoPortal will not treat these as concepts in any of its displays or features. See the W3C's [SKOS System Primer](#) and [SKOS Reference](#) for concept documentation and examples.

Note: Some OWL ontologies declare the SKOS namespace to facilitate minimal use of SKOS constructs for things like labels (e.g., `skos:prefLabel`, `skos:altLabel`) or mappings (e.g., `skos:exactMatch`, `skos:broaderMatch`). In these cases, the proper format for new ontology submissions is OWL, not SKOS.

- **skos:ConceptScheme** and **skos:hasTopConcept.** For every semantic artefact entry in EcoPortal, the application provides a tabbed interface with various views of the semantic artefact data, e.g., a "Classes" tab with a tree structure to graphically depict the hierarchical collection of semantic artefact classes.

In the case of SKOS vocabularies, EcoPortal determines which concepts to display as roots in the concept tree by querying vocabulary content for occurrences of `skos:hasTopConcept` property assertions. Top concepts are the most general concepts contained in SKOS concept schemes (an aggregation of one or more SKOS concepts).

The following example, taken from the SKOS System Primer, shows how to define a concept scheme and link it to the most general concepts it contains:

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#>
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix ex: <http://www.example.com/>
ex:animalThesaurus rdf:type skos:ConceptScheme
skos:hasTopConcept ex:mammals
skos:hasTopConcept ex:fish
```

SKOS vocabularies submitted to EcoPortal must contain a minimum of one concept scheme and top concept assertion. See the [SKOS System Primer](#) and [SKOS Reference](#) for more documentation of concept schemes and top concepts.

If your vocabulary declares more than one concept scheme, all of the top concepts will be aggregated and displayed as root level concepts. The EcoPortal user interface offers functionality to group these top-level concepts based on their respective concept schemes, accessible through the implemented filters.

Hierarchy in SKOS vocabularies

The only semantic relationship in SKOS vocabularies that EcoPortal uses to construct and display concept hierarchies is the `skos:broader` property.

```
ex:mammals rdf:type skos:Concept;
```

```
skos:inScheme :myScheme;
```

```
skos:narrower ex:cat;
```

```
skos:prefLabel 'mammals'@en;
```

```
skos:broader ex:animals.
```

Other properties used to denote hierarchical relationships like `skos:narrower`, `skos:broaderTransitive`, and `skos:narrowerTransitive`, are ignored.

Metrics data for SKOS vocabularies

EcoPortal employs the [OWL API](#) to parse all ontology and vocabulary submissions, including the computation of metric data. The OWL API interprets SKOS vocabularies as RDF files that encompass classes and instances. Following the SKOS Reference, concepts are identified as instances of `owl:Class`, consequently, they are counted as instances, also referred to as "individuals".

When examining [metric tables](#) within the EcoPortal user interface, the "Number Of Individuals" value aligns with the count of concepts within a given SKOS vocabulary.

Example of valid SKOS

This example provides a simple illustration of the composition of a SKOS file that complies with the above constraints.

- **Example header.** The provided header establishes several common namespaces that could prove valuable. The final namespace is the one responsible for defining the specific SKOS vocabulary in question. Ideally, the IRI associated with the "myskosid" namespace should be a resolvable location pointing to the SKOS semantic artefact.

```
<xml version='1.0' encoding='UTF-8'?>
<rdf:RDF
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:rdfs='http://www.w3.org/2000/01/rdf-schema#'
  xmlns:skos='http://www.w3.org/2004/02/skos/core#'
  xmlns:dct='http://purl.org/dc/terms/'
  xmlns:myskosid='https://example.org/ontologies/myskosontology/' >
```

- **Example semantic artefact description.** In the rdf:type item, this namespace is declared as the ConceptScheme. It is important to note that the ConceptScheme doesn't necessarily have to match the semantic artefact namespace. Additional metadata examples are given as good practices in semantic artefact metadata curation. While dct:creator doesn't necessarily need to be an ORCID ID, having a unique identifier is recommended for identifying the creator, whether an individual or organisation. This semantic artefact comprises only two concepts (to be defined later), resulting in only two skos:hasTopConcept declarations.

```
<rdf:Description rdf:about='https://example.org/ontologies/myskosontology/'>
  <rdfs:label xml:lang='en'>Example SKOS ontology for EcoPortal</rdfs:label>
  <rdf:type rdf:resource='http://www.w3.org/2004/02/skos/core#ConceptScheme'/>
  <rdfs:comment xml:lang='en'>Example created to simplify understanding and creation of a SKOS vocab
for EcoPortal</rdfs:comment>
  <dct:created rdf:datatype='http://www.w3.org/2001/XMLSchema#date'>2024-02-02</dct:created>
  <dct:modified rdf:datatype='http://www.w3.org/2001/XMLSchema#date'>2024-02-
02</dct:modified>
  <dct:license rdf:resource='https://creativecommons.org/licenses/by/4.0/'/>
```

```
<dc:creator rdf:resource='https://orcid.org/0000-0001-6875-5360'/>
<skos:hasTopConcept df:resource='https://example.org/ontologies/myskosontology/fragmentid001'/>
<skos:hasTopConcept rdf:resource='https://example.org/ontologies/myskosontology/fragmentid002'/>
</rdf:Description>
```

- **Example term definitions.** This section shows the two concepts and a few typical annotations about those concepts. The first `rdf:Description` line of each group names the concept that is being defined in the indented lines that follows. The `rdf:type` and `skos:prefLabel` are required annotation content for EcoPortal to work effectively. Other items are optional. The `skos:topConceptOf` is not strictly required for EcoPortal SKOS semantic artefacts, but provides useful contextualization if there is more than one `topConcept`.

```
<rdf:Description rdf:about='https://example.org/ontologies/myskosontology/fragmentid001''>
  <rdf:type rdf:resource='http://www.w3.org/2004/02/skos/core#Concept'/>
  <skos:prefLabel xml:lang='en'>First concept</skos:prefLabel>
  <skos:definition xml:lang='en'>The very first example provided as part of this semantic
  artefact.</skos:definition>
  <skos:topConceptOf rdf:resource='https://example.org/ontologies/myskosontology/'/>
</rdf:Description>
<rdf:Description rdf:about='https://example.org/ontologies/myskosontology/fragmentid002''>
  <rdf:type rdf:resource='http://www.w3.org/2004/02/skos/core#Concept'/>
  <skos:prefLabel xml:lang='en'>Second concept</skos:prefLabel>
  <skos:definition xml:lang='en'>The very first example provided as part of this semantic
  artefact.</skos:definition>
  <skos:topConceptOf rdf:resource='https://example.org/ontologies/myskosontology/'/>
</rdf:Description>
```

- **Closing XML.** Needed for a complete, parseable RDF file.

```
</rdf:RDF>
```

5. Developer guide

All the content that the EcoPortal uses (and more!) is available via REST services by means users can:

- access all EcoPortal semantic artefacts, their different versions, and metadata for those versions;
- access information about any semantic artefact concept in EcoPortal (with definitions, synonyms, and other properties);
- search across all semantic artefacts in EcoPortal;
- get hierarchy information for EcoPortal semantic artefacts (such as parents, children, or siblings of a class, roots or leaves of a class hierarchy).

5.1 API key

To use the EcoPortal REST APIs, an API key is required. The API key can be found in the [Account Settings](#) of logged in users (see Figure 46).

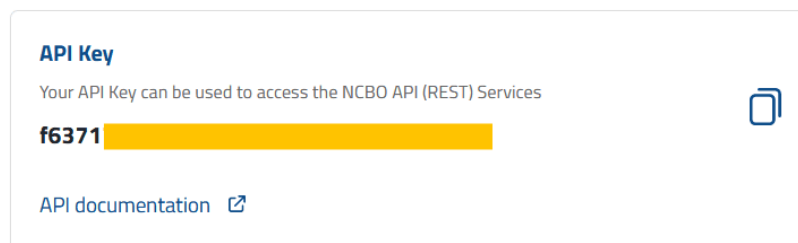


Figure 46: How to retrieve your API key

An API Key is required to access any API call. It can be provided in three ways:

- Using the API key query string parameter

```
#python code example
```

```
import requests
```

```
# example query string parameter on ontology media type:
```

```
# 'https://data.ecoportal.lifewatch.eu/{media types}/{acronym}?apikey=your api key'
```

```
res = requests.get('https://data.ecoportal.lifewatch.eu/ontologies/LUPO?apikey=7404e1c1-XXXX-XXXX-XXXX-XXXXXXXXXXXX')
```

```
Ontology_name = res.json()[0]['name']
```

```
Ontology_id = res.json()['@id']  
print(Ontology_name, Ontology_id)
```

```
result : LifeWatch ERIC Upper Ontology https://data.ecoportal.lifewatch.eu/ontologies/LUPO
```

- Providing an authorization header
Authorization: apikey token=your_apikey

```
#python code example  
import urllib.request, urllib.error, urllib.parse  
import json  
REST_URL = 'https://data.ecoportal.lifewatch.eu'  
API_KEY = 'Your API Key'  
def get_json(url):  
    opener = urllib.request.build_opener()  
    opener.addheaders = [('Authorization', 'apikey token=' + API_KEY)]  
    return json.loads(opener.open(url).read())  
# https://data.ecoportal.lifewatch.eu/{media types}/{acronym}  
resource = get_json(REST_URL + '/ontologies/LUPO')  
# get a desired resource attribute like name and id  
name = resource['name']  
id = resource['@id']  
print(name, id)
```

```
result : LifeWatch ERIC Upper Ontology https://data.ecoportal.lifewatch.eu/ontologies/LUPO
```

- When using a web browser to explore the API, if you provide your API key once using method 1, it will be stored in a cookie for subsequent requests. You can override this by providing a different API key in a new call.

5.2 REST API Documentation

This API consists of a set of resources (Ontologies, Classes, etc) and related endpoints (Search, Annotator, Recommender) that are connected together via links, much like webpages. The developer recommends trying to browse the API using a web browser before starting writing code.

For more information, please contact us at semantics@lifewatch.eu.

5.2.1 Common Parameters

Parameter	Possible Values	Description
include	all {comma-separated list of attributes, EX: attr1,attr2}	By default, the API will show a subset of the available attributes for a given media type. This behaviour can be overridden by providing include=all to show all attributes or include=attribute1,attribute2 to include a specific list. The API is optimised to return the default values, so overriding this can impact the performance of your request. The include=all option is most useful for testing in the browser. Use it to identify the set of attributes required and use only those by passing them as a comma separated list, e.g. include=prefLabel,cui .

		The include parameter is currently unsupported on Annotator and Recommender endpoints.
format	json jsonp xml	The API returns JSON as the default content type. This can be overridden by using the format query string parameter. The API also respects Accept header entries, with precedence given to the format parameter.
page	{integer representing the page number}	For calls that are paged, this will indicate which page number you want to retrieve. You can follow hypermedia links for nextPage and prevPage as well.
pagesize	{integer representing the size of the returned page}	For calls that are paged, this will indicate the size of the page you want to retrieve.
include_views	{boolean representing whether to include ontology views (default is false)}	For calls that involve ontologies, include_views='true' will include ontology views.
display_context	{true false} (defaults to true)	Turn off the JSON-LD context serialisation. This will reduce the response size significantly for some calls, speeding up transmission and parse

		time.
display_links	{true false} (defaults to true)	Turn off the hypermedia link serialisation. This will reduce the response size significantly for some calls, speeding up transmission and parse time.
download_format	{csv rdf} (defaults to user-uploaded file format)	Allows you to specify alternative formats for ontology file downloads for ontology and ontology submission download endpoints. CSV is only available for the most recent submission.

5.2.2 Resource Endpoints

The EcoPortal API endpoints include a dictionary named "links" containing URLs of the available media types within the API along with hypermedia links connecting them. These media types describe the various types of resources available, specifying the HTTP verbs (POST, GET, PUT, DELETE, and PATCH) that can be used with them and outlining the attributes contained within each resource.

Available Media Types:

- **agents:** <https://data.ecoportal.lifewatch.eu/Agents>
- **categories:** <https://data.ecoportal.lifewatch.eu/categories>
- **groups:** <https://data.ecoportal.lifewatch.eu/groups>
- **Class:** <https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes>
- **documentation:** <https://data.ecoportal.lifewatch.eu/documentation>
- **identifier_requests:** https://data.ecoportal.lifewatch.eu/identifier_requests
- **mappings:** <https://data.ecoportal.lifewatch.eu/mappings>
- **metrics:** <https://data.ecoportal.lifewatch.eu/metrics>
- **notes:** <https://data.ecoportal.lifewatch.eu/notes>

- **ontologies:** <https://data.ecoportal.lifewatch.eu/ontologies>
- **ontologies_full:** https://data.ecoportal.lifewatch.eu/ontologies_full
- **analytics:** <https://data.ecoportal.lifewatch.eu/analytics>
- **submissions:** <https://data.ecoportal.lifewatch.eu/submissions>
- **projects:** <https://data.ecoportal.lifewatch.eu/projects>
- **provisional_classes:** https://data.ecoportal.lifewatch.eu/provisional_classes
- **provisional_relations:** https://data.ecoportal.lifewatch.eu/provisional_relations
- **replies:** <https://data.ecoportal.lifewatch.eu/replies>
- **reviews:** <https://data.ecoportal.lifewatch.eu/reviews>
- **slices:** <https://data.ecoportal.lifewatch.eu/slices>
- **submission_metadata:** https://data.ecoportal.lifewatch.eu/submission_metadata
- **ontology_metadata:** https://data.ecoportal.lifewatch.eu/ontology_metadata
- **users:** <https://data.ecoportal.lifewatch.eu/users>

Agents

Example: <https://data.ecoportal.lifewatch.eu/Agents>

HTTP Methods for Resource

HTTP Verb	Path
GET	/agents
GET	/agents/:id
GET	/Agents
GET	/Agents/:id
PUT	/agents/:acronym
PUT	/Agents/:acronym
POST	/agents
POST	/Agents
PATCH	/agents/:id
PATCH	/Agents/:id

DELETE	/agents/:id
DELETE	/Agents/:id

Resource Description

Attribute	Default	Unique	Required	List	Type
agentType	true		true		
name	true		true		
homepage	true				
acronym	true				
email	true	true			
identifiers	true			true	http://www.w3.org/ns/adms#Identifier
affiliations	true			true	http://xmlns.com/foaf/0.1/Agent
creator	true		true		https://data.ecoportal.lifewatch.eu/metadata/users

Categories

Example: <https://data.ecoportal.lifewatch.eu/categories>

Example: <https://data.ecoportal.lifewatch.eu/categories/Ecology>

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/categories>

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies/:acronym/categories
GET	/categories
GET	/categories/:acronym

PUT	/categories/:acronym
POST	/categories
PATCH	/categories/:acronym
DELETE	/categories/:acronym

Resource Description

Attribute	Default	Unique	Required	List	Type
acronym	true	true	true		
name	true		true		
description	true				
created	true				
parentCategory	true				https://data.ecoportal.lifewatch.eu/metadata/Categories
ontologies	true				https://data.ecoportal.lifewatch.eu/metadata/ontology

Class

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes>

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/roots>

Example:
https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_2759

Example:
https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_2759/children

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies/:ontology/classes
GET	/ontologies/:ontology/classes/roots_paged
GET	/ontologies/:ontology/classes/roots
GET	/ontologies/:ontology/classes/:cls
GET	/ontologies/:ontology/classes/:cls/paths_to_root
GET	/ontologies/:ontology/classes/:cls/tree
GET	/ontologies/:ontology/classes/:cls/ancestors
GET	/ontologies/:ontology/classes/:cls/descendants
GET	/ontologies/:ontology/classes/:cls/children
GET	/ontologies/:ontology/classes/:cls/parents

Resource Description

Attribute	Default	Unique	Required	List	Type
prefLabel	true		true		
synonym	true			true	
definition	true			true	
obsolete	true				
semanticType	true			true	
cui	true			true	
inScheme	true			true	
memberOf	true				http://www.w3.org/2004/02/skos/core#Collection
created	true				

modified	true				
label				true	
prefLabelXI				true	http://www.w3.org/2008/05/skos-xl#Label
altLabelXI				true	http://www.w3.org/2008/05/skos-xl#Label
hiddenLabelXI				true	http://www.w3.org/2008/05/skos-xl#Label
notation					
prefixIRI					
parents				true	http://www.w3.org/2002/07/owl#Class
ancestors					http://www.w3.org/2002/07/owl#Class
children					http://www.w3.org/2002/07/owl#Class
subClassOf				true	
xref					
notes					https://data.ecoportallifewatch.eu/metadata/Note
properties					
childrenCount					
hasChildren					

Related Hypermedia Links

Type	URL
Self	https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_131567
ontology	https://data.ecoportal.lifewatch.eu/ontologies/PCO
children	https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_131567/children
parents	https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_131567/parents
descendants	https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_131567/descendants
ancestors	https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_131567/ancestors
instances	https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_131567/instances
tree	https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_131567/tree
notes	https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_131567/notes
mappings	https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_131567/mappings

Groups

Example: <https://data.ecoportal.lifewatch.eu/groups>

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/groups>

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies/:acronym/groups
GET	/groups
GET	/groups/:acronym
PUT	/groups/:acronym
POST	/groups
PATCH	/groups/:acronym
DELETE	/groups/:acronym

Resource Description

Attribute	Default	Unique	Required	List	Type
acronym	true	true	true		
name	true		true		
description	true				
created	true				
ontologies	true				https://data.ecoportal.lifewatch.eu/metadata/Ontologies

Identifier requests

Example: https://data.ecoportal.lifewatch.eu/identifier_requests

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies/:acronym/identifier_requests
GET	/identifier_requests

GET	/identifier_requests/all_doi_requests
GET	/identifier_requests/:requestId
GET	/identifier_requests/:requestId/submission
GET	/identifier_requests/:requestId/requestedBy
GET	/identifier_requests/:requestId/processedBy
POST	/identifier_requests
PATCH	/identifier_requests/:requestId
DELETE	/identifier_requests/all
DELETE	/identifier_requests/:requestId

Resource Description

Attribute	Default	Unique	Required	List	Type
requestId	true	true	true		
status	true		true		
requestType	true		true		
requestedBy	true		true		https://data.ecoportal.lifewatch.eu/metadata/users
requestDate	true		true		
processedBy	true				https://data.ecoportal.lifewatch.eu/metadata/users
processingDate	true				
message	true				
submission	true				https://data.ecoportal.lifewatch.eu/metadata/OntologySubmission

Named Individual

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/instances>

Example:
https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_2759/instances

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies/:ontology/classes/:cls/instances
GET	/ontologies/:ontology/instances
GET	/ontologies/:ontology/instances/:inst

Resource Description

Attribute	Default	Unique	Required	List	Type
label	true	true		true	
prefLabel	true		true		
types	true			true	
properties	true				

Mapping

Example: <https://data.ecoportal.lifewatch.eu/mappings>

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/mappings>

Example: <https://data.ecoportal.lifewatch.eu/mappings/statistics/ontologies>

Example:
https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FNCBITaxon_2759/mappings

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies/:ontology/classes/:cls/mappings
GET	/ontologies/:ontology/mappings
GET	/mappings
GET	/mappings/recent
GET	/mappings/:mapping
GET	/mappings/statistics/ontologies
GET	/mappings/statistics/ontologies/:ontology
GET	/mappings/statistics/interportal/:ontology
GET	/mappings/statistics/external
POST	/mappings
POST	/mappings/load
PATCH	/mappings/:mapping
DELETE	/mappings/:mapping

Resource Description

Attribute	Default	Unique	Required	List	Type
process					
source					
classes					
id					

Metrics

Example: <https://data.ecoportal.lifewatch.eu/metrics>

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/metrics>

HTTP Methods for Resource

HTTP Verb	Path
GET	/metrics
GET	/metrics/missing
GET	/ontologies/:ontology/metrics
GET	/ontologies/:ontology/submissions/:ontology_submission_id/metrics

Resource Description

Attribute	Default	Unique	Required	List	Type
submission	true				https://data.ecoportal.lifewatch.eu/metadata/OntologySubmission
created	true				
classes	true		true		
individuals	true		true		
properties	true		true		
maxDepth	true		true		
maxChildCount	true		true		
averageChildCount	true		true		
classesWithOneChild	true		true		
classesWithMoreThan25Children	true		true		
classesWithNo	true		true		

Definition					
numberOfAxioms	true				
entities	true				

Note

Example: <https://data.ecoportal.lifewatch.eu/notes>

Example: <https://data.ecoportal.lifewatch.eu/notes/aca5bcc0-e90b-0138-2438-000c29c082c0>

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies/:ontology/notes?:include_threads?
GET	/ontologies/:ontology/classes/:cls/notes?:include_threads?
GET	/notes?:include_threads?
GET	/notes/:noteid?:include_threads?
POST	/notes
PATCH	/notes/:noteid
DELETE	/notes/:noteid

Resource Description

Attribute	Default	Unique	Required	List	Type
subject	true				
body	true				
creator	true		true		https://data.ecoportal.lifewatch.eu/metadata/users
created	true				

archived	true				
createdInSubmission	true				https://data.ecoportal.lifewatch.eu/metadata/OntologySubmissions
reply	true			true	
relatedOntology	true		true	true	
relatedClass	true			true	http://www.w3.org/2002/07/owl#Class
proposal	true				

Reply

Example: <https://data.ecoportal.lifewatch.eu/replies>

HTTP Methods for Resource

HTTP Verb	Path
GET	/notes/:noteid/replies
GET	/replies?:include_threads?
GET	/replies/:replyid
POST	/replies
PATCH	/replies/:replyid
DELETE	/replies/:replyid

Resource Description

Attribute	Default	Unique	Required	List	Type
body	true		true		
creator	true		true		https://data.ecoportal.lifewatch.eu/metadata/users

created	true				
children	true				
parent					https://data.ecoportal.lifewatch.eu/metadata/User

Object Property

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/properties>

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/properties/roots>

Example:
https://data.ecoportal.lifewatch.eu/ontologies/PCO/properties/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FRO_0002500

Example:
https://data.ecoportal.lifewatch.eu/ontologies/PCO/properties/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FRO_0002500/label

Example:
https://data.ecoportal.lifewatch.eu/ontologies/PCO/properties/http%3A%2F%2Fpurl.obolibrary.org%2Fobo%2FRO_0002500/children

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies/:ontology/properties
GET	/ontologies/:ontology/properties/roots
GET	/ontologies/:ontology/properties/:property
GET	/ontologies/:ontology/properties/:property/label
GET	/ontologies/:ontology/properties/:property/tree
GET	/ontologies/:ontology/properties/:property/ancestors
GET	/ontologies/:ontology/properties/:property/descendants
GET	/ontologies/:ontology/properties/:property/parents
GET	/ontologies/:ontology/properties/:property/children

Resource Description

Attribute	Default	Unique	Required	List	Type
label	true			true	
definition	true			true	
parents	true			true	http://www.w3.org/2002/07/owl#ObjectProperty
children	true				http://www.w3.org/2002/07/owl#ObjectProperty
submission					
ancestors					
descendants					

Ontology

Example: <https://data.ecoportal.lifewatch.eu/ontologies>

Example: https://data.ecoportal.lifewatch.eu/ontologies_full

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO>

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/download>

Example: https://data.ecoportal.lifewatch.eu/ontologies/latest_submission

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies
GET	/ontologies/:acronym
GET	/ontologies/:acronym/latest_submission
GET	/ontologies/:acronym/latest_submission/datacite_metadata_json
GET	/ontologies/:acronym/latest_submission/ecoportal_metadata_json

GET	/ontologies/:acronym/download
GET	/ontologies_full
PUT	/ontologies/:acronym
POST	/ontologies
PATCH	/ontologies/:acronym/latest_submission
PATCH	/ontologies/:acronym
DELETE	/ontologies/:acronym

Resource Description

Attribute	Default	Unique	Required	List	Type
acronym	true	true	true		
name	true	true	true		
administeredBy	true		true	true	https://data.ecoportal.lifewatch.eu/metadata/User
group	true			true	https://data.ecoportal.lifewatch.eu/metadata/Group
viewingRestriction	true				
flat	true				
hasDomain	true			true	https://data.ecoportal.lifewatch.eu/metadata/Category
summaryOnly	true				
viewOf	true				https://data.ecoportal.lifewatch.eu/metadata/Ontology
views	true				https://data.ecoportal.lifewatch.eu/metadata/Ontology
ontologyType	true				
					https://data.ecoportal.lifewatch.eu/metadata/Ontology

submissions					h.eu/metadata/OntologySubmission
projects					https://data.ecoportal.lifewatch.eu/metadata/Project
notes					https://data.ecoportal.lifewatch.eu/metadata/Note
reviews					https://data.ecoportal.lifewatch.eu/metadata/Review
provisionalClasses					
subscriptions					https://data.ecoportal.lifewatch.eu/metadata/OntologySubmissions
doNotUpdate					
acl				true	http://data.bioontology.org/metadata/User

Related Hypermedia Links

Type	URL
submissions	https://data.ecoportal.lifewatch.eu/ontologies/PCO/submissions
properties	https://data.ecoportal.lifewatch.eu/ontologies/PCO/properties
classes	https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes
single_class	https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/{class_id}
roots	https://data.ecoportal.lifewatch.eu/ontologies/PCO/classes/roots
schemes	https://data.ecoportal.lifewatch.eu/ontologies/PCO/schemes
collections	https://data.ecoportal.lifewatch.eu/ontologies/PCO/collections
xl_labels	https://data.ecoportal.lifewatch.eu/ontologies/PCO/skos_xl_lab

	els
reviews	https://data.ecoportal.lifewatch.eu/ontologies/PCO/instances
metrics	https://data.ecoportal.lifewatch.eu/ontologies/PCO/metrics
reviews	https://data.ecoportal.lifewatch.eu/ontologies/PCO/reviews
notes	https://data.ecoportal.lifewatch.eu/ontologies/PCO/notes
groups	https://data.ecoportal.lifewatch.eu/ontologies/PCO/groups
categories	https://data.ecoportal.lifewatch.eu/ontologies/PCO/categories
latest_submission	https://data.ecoportal.lifewatch.eu/ontologies/PCO/latest_submission
projects	https://data.ecoportal.lifewatch.eu/ontologies/PCO/projects
download	https://data.ecoportal.lifewatch.eu/ontologies/PCO/download
views	https://data.ecoportal.lifewatch.eu/ontologies/PCO/views
analytics	https://data.ecoportal.lifewatch.eu/ontologies/PCO/analytics
ui	http://ecoportal.lifewatch.eu/ontologies/PCO
identifier_requests	https://data.ecoportal.lifewatch.eu/ontologies/PCO/identifier_requests

Ontology Submission

Example: <https://data.ecoportal.lifewatch.eu/submissions>

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/submissions>

HTTP Methods for Resource

HTTP Verb	Path
GET	/submissions
GET	/ontologies/:acronym/submissions
GET	/ontologies/:acronym/submissions/:ontology_submission_id

GET	/ontologies/:acronym/submissions/:ontology_submission_id/datacite_metadata_json
GET	/ontologies/:acronym/submissions/:ontology_submission_id/download
GET	/ontologies/:acronym/submissions/:ontology_submission_id/download_diff
POST	/submissions
POST	/ontologies/:acronym/submissions
PATCH	/ontologies/:acronym/submissions/:ontology_submission_id
DELETE	/ontologies/:acronym/submissions/:ontology_submission_id

Resource Description

Attribute	Default	Unique	Required	List	Type
submissionId	true		true		
version	true				
status	true		true		
hasOntologyLanguage	true		true		
description	true		true		
homepage	true				
documentation	true				
publication	true		true		
released	true		true		
creationDate	true				
contact	true		true	true	
ontology	true		true		https://data.ecoportallifewatch.eu/metadata/Ontology

prefLabelProperty					
definitionProperty					
synonymProperty					
authorProperty					
classType					
hierarchyProperty					
obsoleteProperty					
obsoleteParent					
createdProperty					
modifiedProperty					
URI			true		
versionIRI					
deprecated					
hasFormalityLevel					
hasOntologySyntax					
naturalLanguage				true	
isOfType				true	

identifier				true	
notes				true	
keywords				true	
hiddenLabel				true	
alternative				true	
abstract					
hasLicense					
useGuidelines					
morePermissions					
copyrightHolder					http://xmlns.com/foaf/0.1/Agent
valid					
curatedOn				true	
modificationDate					
hasCreator				true	http://xmlns.com/foaf/0.1/Agent
hasContributor				true	http://xmlns.com/foaf/0.1/Agent
curatedBy				true	http://xmlns.com/foaf/0.1/Agent
publisher				true	http://xmlns.com/foaf/0.1/Agent
fundedBy				true	http://xmlns.com/foaf/0.1/Agent
endorsedBy				true	
translator				true	http://xmlns.com/foaf/0.1/Agent

					ent
audience					
repository					
bugDatabase					
mailingList					
toDoList				true	
award				true	
knownUsage				true	
designedForOntologyTask				true	
hasDomain				true	
coverage					
example				true	
conformsToKnowledgeRepresentationParadigm					
usedOntologyEngineeringMethodology					
usedOntologyEngineeringTool				true	
accrualMethod				true	
accrualPeriodicity					
accrualPolicy					
competencyQu				true	

estion					
wasGenerated By				true	
wasInvalidated By				true	
pullLocation					
isFormatOf					
hasFormat				true	
dataDump					
csvDump					
uriLookupEndp oint					
openSearchDe scription					
source				true	
endpoint				true	
includedInData Catalog				true	
hasPriorVersio n					
hasPart				true	
ontologyRelat edTo				true	
similarTo				true	
comesFromTh eSameDomain				true	
isAlignedTo				true	
isBackwardCo				true	

mpatibleWith					
isIncompatible With				true	
hasDisparate Modelling				true	
hasDisjunction sWith				true	
generalizes				true	
explanationEv olution				true	
useImports				true	
usedBy				true	
workTranslatio n				true	
translationOf Work				true	
uriRegexPatter n					
preferredNam espaceUri					
preferredNam espacePrefix					
exampleIdentif ier					http://www.w3.org/2002/07/owl#Class
keyClasses				true	http://www.w3.org/2002/07/owl#Class
metadataVoc				true	
uploadFilePath					
diffFilePath					

masterFileName					
associatedMedia				true	
depiction				true	
logo					
metrics					https://data.ecoportal.lifewatch.eu/metadata/Metrics
submissionStatus				true	
missingImports				true	
identifierRequests					https://data.ecoportal.lifewatch.eu/metadata/IdentifierRequest

Project

Example: <https://data.ecoportal.lifewatch.eu/projects>

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/projects>

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies/:acronym/projects
GET	/projects
GET	/projects/:acronym
PUT	/projects/:acronym
POST	/projects/
PATCH	/projects/:acronym
DELETE	/projects/:acronym

Resource Description

Attribute	Default	Unique	Required	List	Type
acronym	true	true	true		
creator	true		true	true	https://data.ecoportal.lifewatch.eu/metadata/Users
created	true				
updated	true				
name	true		true		
homePage	true		true		
description	true		true		
contacts	true				
institution	true				
ontologyUsed	true			true	https://data.ecoportal.lifewatch.eu/metadata/Ontology

Provisional Class

Example: https://data.ecoportal.lifewatch.eu/provisional_classes

Example: https://data.ecoportal.lifewatch.eu/ontologies/PCO/provisional_classes

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies/:ontology/provisional_classes
GET	/users/:user/provisional_classes
GET	/provisional_classes
GET	/provisional_classes/:provisional_class_id

POST	/provisional_classes
PATCH	/provisional_classes/:provisional_class_id
DELETE	/provisional_classes/:provisional_class_id

Resource Description

Attribute	Default	Unique	Required	List	Type
label	true		true		
synonym	true			true	
definition	true			true	
subclassOf	true				
creator	true		true		https://data.ecoportal.lifewatch.eu/metadata/Users
created	true				
permanentId	true				
notId	true				
ontology	true				http://data.bioontology.org/metadata/Ontology
relations	true				

Provisional Relation

Example: https://data.ecoportal.lifewatch.eu/provisional_relations

HTTP Methods for Resource

HTTP Verb	Path
GET	/provisional_relations
GET	/provisional_relations/:provisional_relation_id
POST	/provisional_relations

DELETE	/provisional_relations/:provisional_relation_id
--------	---

Resource Description

Attribute	Default	Unique	Required	List	Type
source	true		true		http://data.bioontology.org/metadata/ProvisionalClass
relationType	true		true		
targetClassId	true		true		
targetClassOntology	true		true		http://data.bioontology.org/metadata/Ontology
creator	true		true		http://data.bioontology.org/metadata/User
created	true				

Review

Example: <https://data.ecoportal.lifewatch.eu/reviews>

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/reviews>

This endpoint is deprecated.

Collection

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/collections>

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies/:ontology/collections
GET	/ontologies/:ontology/collections/:collection
GET	/ontologies/:ontology/collections/:collection/members

Resource Description

Attribute	Default	Unique	Required	List	Type
prefLabel	true		true		
memberCount	true				
properties					

Concept Scheme

Example: <https://data.ecoportal.lifewatch.eu/ontologies/PCO/schemes>

HTTP Methods for Resource

HTTP Verb	Path
GET	/ontologies/:ontology/schemes
GET	/ontologies/:ontology/schemes/:scheme

Resource Description

Attribute	Default	Unique	Required	List	Type
prefLabel	true		true		
properties	true				

Slice

Example: <https://data.ecoportal.lifewatch.eu/slices>

HTTP Methods for Resource

HTTP Verb	Path
GET	/slices
GET	/slices/synchronize_groups
GET	/slices/:slice_id
POST	/slices

PATCH	/slices/:slice
DELETE	/slices/:slice

Resource Description

Attribute	Default	Unique	Required	List	Type
acronym	true	true	true		
name	true		true		
description	true				
created	true				
ontologies	true		true	true	https://data.ecoportal.lifewatch.eu/metadata/Ontology

User

Example: <https://data.ecoportal.lifewatch.eu/users>

HTTP Methods for Resource

HTTP Verb	Path
GET	/users
GET	/users/:username
PUT	/users/:username
POST	/users/:authenticate
POST	/users/:create_reset_password_token
POST	/users/:reset_password
POST	/users
PATCH	/users/:username
DELETE	/users/:username

5.2.3 Service endpoints

Multiple endpoints are provided for conducting lookups pertaining to classes, annotations, and annotated resources. Although these endpoints do not strictly adhere to REST principles, they do return objects containing relevant links when applicable.

5.2.3.1 Term Search

`https://data.ecoportal.lifewatch.eu/search?q={search query}`

Example: <https://data.ecoportal.lifewatch.eu/search?q=Eukaryota>

#python code example

```
import urllib.request, urllib.error, urllib.parse
import json
```

```
REST_URL = 'https://data.ecoportal.lifewatch.eu'
```

```
API_KEY = 'API_KEY = 'Your API Key''
```

```
def get_json(url):
```

```
    opener = urllib.request.build_opener()
```

```
    opener.addheaders = [('Authorization', 'apikey token=' + API_KEY)]
```

```
    return json.loads(opener.open(url).read())
```

```
# define a list of terms
```

```
terms_list = ['heart', 'lung', 'experiment', 'human', 'brain', 'melanoma']
```

```
# Search for terms
```

```
search_results = []
```

```
for term in terms_list:
```

```
    search_results.append(get_json(REST_URL + '/search?q=' + term)['collection'])
```

The same endpoint also allows limiting a search to a given subtree/branch. The root of the subtree is defined using a combination of parameters "ontology" and

"subtree_root_id". Both are required when performing a subtree search.

Parameters :

- `ontologies={ontology_id1,ontology_id2,ontology_id3}`
- `require_exact_match={true|false}` // default = false
- `suggest={true|false}` // default = false. Will perform a search specifically geared towards type-ahead suggestions.
- `also_search_views={true|false}` // Include ontology views in the search. default = false
- `require_definitions={true|false}` // default = false
- `also_search_properties={true|false}` // default = false
- `also_search_obsolete={true|false}` // default = false (exclude obsolete terms)
- `cui={C0018787,C0225807,C0018787}` // Allows restricting query by CUIs. default = null (no restriction)
- `semantic_types={T023,T185,T061}` // Allows restricting query by Semantic Types (TUI). default = null (no restriction)
- `include={prefLabel, synonym, definition, notation, cui, semanticType}` // default = (see Common Parameters section)
- `page={integer representing the page number}` // default = 1
- `pagesize={integer representing the size of the returned page}` // default = 50

Subtree Search:

The same endpoint also allows limiting a search to a given subtree/branch. The root of the subtree is defined using a combination of parameters "**ontology**" and "**subtree_root_id**". Both are required when performing a subtree search.

`https://data.ecoportal.lifewatch.eu/search?q={searchquery}&ontology={acronym}&subtree_root_id={a URI encoded class id of the root node of the subtree}`

Roots Only Search :

The same endpoint allows limiting a search to root classes only.

Example:

https://data.ecoportal.lifewatch.eu/search?q=Eukaryota&ontologies=PCO&roots_only=true

5.2.3.2 Ontology Property Search

https://data.ecoportal.lifewatch.eu/property_search?q={search query}

Example: https://data.ecoportal.lifewatch.eu/property_search?q=hasPart

An endpoint for searching ontology properties by their labels and IDs (URIs)

Parameters :

- `ontologies={ontology_id1,ontology_id2,ontology_id3}`
- `require_exact_match={true|false}` // Restricts results only to the exact matches of the query by the property id, label, or the generated label (a label, auto-generated from the ID). default = false
- `also_search_views={true|false}` // Include ontology views in the search. default = false
- `require_definitions={true|false}` // Filter results only to those that include definitions. default = false
- `include={label, labelGenerated, definition, parents}` // default = All of the properties in this example
- `ontology_types={ONTOLOGY,VALUE_SET_COLLECTION}` // default = all ontology types are returned
- `property_types={object,annotation,datatype}` // Restricts results to specific property types (object, annotation, datatype). default = all property types are returned
- `page={integer representing the page number}` // default = 1
- `pagesize={integer representing the size of the returned page}` // default = 50

5.2.3.3 Annotator

<https://data.ecoportal.lifewatch.eu/annotator?text={input text}>

Example:

<https://data.ecoportal.lifewatch.eu/annotator?text=a+principal+taxonomic+category+that+ranks+above+class+and+below+kingdom+equivalent+to+the+division+in+botany>

The Annotator endpoint will examine text input and return relevant classes.

Parameters :

- `ontologies={ontology_id1,ontology_id2..,ontology_idN}`
- `semantic_types={semType1,semType2..,semTypeN}`
- `expand_semantic_types_hierarchy={true|false}` // default = false. true means to use the semantic types passed in the "semantic_types" parameter as well as all their immediate children. false means to use ONLY the semantic types passed in the "semantic_types" parameter.
- `expand_class_hierarchy={true|false}` // default = false. used only in conjunction with "class_hierarchy_max_level" parameter; determines whether or not to include ancestors of the given class when performing an annotation.
- `class_hierarchy_max_level={0..N}` // default = 0. the depth of the hierarchy to use when performing an annotation.
- `expand_mappings={true|false}` // default = false. true means that the following manual mappings will be used in annotation: UMLS, REST, CUI, OBOXREF.
- `stop_words={word1,word2..,wordN}` (case insensitive)
- `minimum_match_length={0..N}`
- `exclude_numbers={true|false}` // default = false
- `whole_word_only={true|false}` // default = true
- `exclude_synonyms={true|false}` // default = false
- `longest_only={true|false}` // default = false. true means that only the longest match for a given phrase will be returned.

Default stop words

The following stop words are used by default:

I, a, above, after, against, all, alone, always, am, amount, an, and, any, are, around, as, at, back, be, before, behind, below, between, bill, both, bottom, by, call, can, co, con, de, detail, do, done, down, due, during, each, eg, eight, eleven, empty, ever, every, few, fill, find, fire, first, five, for, former, four, from, front, full, further, get, give, go, had, has, hasnt, he, her, hers, him, his, i, ie, if, in, into, is, it, last, less, ltd, many, may, me, mill, mine, more, most, mostly, must, my, name, next, nine, no, none, nor, not, nothing, now, of, off, often, on, once, one, only, or, other, others, out, over, part, per, put, re, same, see, serious, several, she, show, side, since, six, so, some, sometimes, still, take, ten, the, then, third, this, thick, thin, three, through, to, together, top, toward, towards, twelve, two, un, under, until, up,

upon, us, very, via, was, we, well, when, while, who, whole, will, with, within, without, you, yourself, yourselves

5.2.3.4 Recommender

<https://data.ecoportal.lifewatch.eu/recommender?input={input text or keywords}>

Example:

<https://data.ecoportal.lifewatch.eu/recommender?input=a+principal+taxonomic+category.+that+ranks+above+class+and+below+kingdom+equivalent+to+the+division+in+botany>

The Recommender endpoint receives a text or a list of keywords and suggests appropriate ontologies for them.

Parameters

- `input_type={1|2}` // default = 1. 1 means that the input type is text. 2 means that the input type is a list of comma separated keywords.
- `output_type={1|2}` // default = 1. 1 means that the output will be a ranked list of individual ontologies. 2 means that the output will be a ranked list of ontology sets.
- `max_elements_set={2|3|4}` // default = 3. Maximum number of ontologies per set (only for `output_type = 2`).
- `wc={value in the range [0,1]}` // default = 0.55. Weight assigned to the ontology coverage criterion.
- `wa={value in the range [0,1]}` // default = 0.15. Weight assigned to the ontology acceptance criterion.
- `wd={value in the range [0,1]}` // default = 0.15. Weight assigned to the ontology detail criterion.
- `ws={value in the range [0,1]}` // default = 0.15. Weight assigned to the ontology specialisation criterion.
- `ontologies={ontology_id1, ontology_id2, ..., ontology_idN}` // default = (empty) (all BioPortal ontologies will be evaluated).

Batch

<https://data.ecoportal.lifewatch.eu/batch>

In order to reduce the number of HTTP requests required to obtain information about resources, we are exploring the use of batch endpoints that will use one request/response cycle for multiple resources. Currently, only classes are supported.

The batch service takes input in the BODY of a POST request. The input should be a JSON document with the following structure:

```
{
  'http://www.w3.org/2002/07/owl#Class': {
    'collection': [
      {
        'class': 'http://example.org/ontology1/class1',
        'ontology': 'http://example.org/ontology1'
      },
      {
        'class': 'http://example.org/ontology2/class1',
        'ontology': 'http://example.org/ontology2'
      },
      {
        'class': 'http://example.org/ontologyN/classN',
        'ontology': 'http://example.org/ontologyN'
      }
    ],
    'display': 'prefLabel,synonym,semanticTypes'
  }
}
```

5.2.3.5 Analytics

<https://data.ecoportal.lifewatch.eu/analytics>

Allows to retrieve analytics for all ontologies for a specific month/year combination.

Get analytics data for a given month/year:

<https://data.ecoportal.lifewatch.eu/analytics?month=4&year=2021>

Get analytics data for a given ontology:

<http://data.ecoportal.lifewatch.eu/ontologies/PCO/analytics>

5.3 Using EcoPortal technology in your project

Several ways exist in which EcoPortal technology can be applied. Here are some use cases:

- **I have a list of terms that were used to annotate a dataset, how can I see if any of these terms exist in standard semantic artefacts?** Use the "Search" web service to search terms in all EcoPortal semantic artefacts. If you have a limited number of terms, searching each one individually via the EcoPortal Search tool with the user interface may be sufficient. Otherwise, for a long list of terms, use the "Search" web service directly.
- **I would like to find all the terms mapped from my semantic artefact to other semantic artefacts in EcoPortal.** Use the "Mapping" web service to find terms in other EcoPortal semantic artefacts that are similar to the terms in your semantic artefact. These mappings include those generated manually (**DbXrefs**) and programmatically (**LOOM**).
- **I have a web form including text fields which I would like to populate with semantic artefact terms, e.g., to select relevant terms from a specific EcoPortal semantic artefact.** Use the "Autocomplete" widget. Click on the "Get Code" button for the code and instructions on how to add this JavaScript to your web form.
- **I would like to add the graph displaying my semantic artefact on my website.** Use the "Visualization" widget. Click on the "Get Code" button for the code and instructions on how to add this JavaScript to your website.
- **I would like to get all terms from a semantic artefact and re-format the data to use in my own system.** Use the "Get All Terms" web service to get all the terms and properties for the semantic artefact that you are interested in.
- **I need to navigate through a semantic artefact and go through the semantic artefact hierarchy for my software application.** Is there a web service to do this? Use the "Get All terms" web service.
- **How can I limit the selection of semantic artefacts used with the Annotator to a given Category or Domain?** Use the "List all semantic artefacts" and "Get Groups/Domains" web service to identify the group/domain of interest.

5.4 EcoPortal widgets: how to use them

EcoPortal widgets are snippets of HTML or JavaScript code that users can embed in their website/web form to use the EcoPortal functionalities. Using EcoPortal widgets is just one of the ways in which users can use EcoPortal technologies on your website/application.

Developers can add any EcoPortal widget to their HTML page or web form by following these steps:

- find the semantic artefact that users are interested in by browsing the list of [EcoPortal semantic artefacts](#) (e.g., Fish Traits Thesaurus);
- click on the semantic artefact name to get to the page with [Details for that semantic artefact](#) (e.g., the details for Fish Traits Thesaurus);
- in the semantic artefact details page, go to the "Widgets" tab;
- select the widget that you want to use;
- click the button below the widget to get usage instructions.

Term-selection field on a form

Developers can add a text field to their web form that will allow users to enter a term from a controlled vocabulary (e.g., terms from a single semantic artefact)

Example use case: suppose that a developer is running a tissue microarray database and that users can upload sample descriptions to the database by means of a web form. Usually, there is a field to specify the diagnosis related to the tissue sample that the user is filling in. This field is often a text-box or a dropdown menu, populated with a list of controlled terms. The free text-box is prone to errors, while the dropdown becomes unmanageable with too many terminologies. The EcoPortal term-selection widget allows users to easily select a term from an ontology or a controlled vocabulary to fill in the specific field. For example, when the user starts typing "fish", the term "fish traits" pops up.

What does the term-selection field do?

- Auto-complete the text field so that users don't need to type the whole term.
- The semantic artefact provides consistency in the manner in which different users use a specific term.

- When a new version of your semantic artefact becomes available in EcoPortal, the widget will automatically use that new version.

To add a field to a web form that allows users to retrieve a term from the semantic artefact of interest, do the following:

- download the [form_complete.js](#) file and put it on your server;
- in the header of the page where you want the field, include the `form_complete.js` file;
- in the form, specify the field class in the following format:

`bp_form_complete-{semantic_resource_acronym_list}-{value}`

- **Example:** `bp_form_complete-FISHTRAITS-uri` will use FISH TRAITS Thesaurus (the acronym is FISHTRAITS) and will put the term URI in the field after the user selects the term from the pull-down list

- **Note:** In addition to single ontology IDs, you can use a list:

`bp_form_complete-FISHTRAITS, ENVTHES-uri` OR you can use 'all' to search across all EcoPortal semantic artefact: `bp_form_complete-all-uri`.

- Developers can use the following parameters to select which value will be placed into the user-visible input field:

- **uri:** the complete URI of the term (e.g., `http://thesauri.lifewatchitaly.eu/FishTraits/?tema=5`);
- **shortid:** the short ID of the term, as used in EcoPortal (e.g., `?tema=5`);
- **name:** the preferred name of the term (e.g., "Fish Trait").

Auto-generated hidden form elements

In addition to the input element you defined, there are four hidden form elements that are automatically created and set when users select a term from the list. For example, if you create a field with this code:

```
<input type='text' name='a' class='bp_form_complete-all-uri' size='100'/>
```

The 'name' attribute is used to create the four following fields (note how the 'a' from the name attribute is appended to the id attribute:

```
<input type='hidden' id='a_ecoportal_preferred_name'>
```

```
<input type='hidden' id='a_ecoportal_concept_id'>
```

```
<input type='hidden' id='a_ecoportal_ontology_id'>
```

```
<input type='hidden' id='a_ecoportal_full_id'>
```

Optional attributes

Include term definitions. Use `data-bp_include_definitions='true'` to display definitions with the list of terms that are returned. For example, use this code to display search results for 'heart' terms, with definitions, in the FISHTRAITS Thesaurus:

```
<input type='text' name='m' data-bp_include_definitions='true' class='bp_form_complete-FISHTRAITS-name' />
```

Limit search to terms below a root term. Use `data-bp_search_branch='{class_id}'`, replacing {class_id} with the class ID for the term to use as the branch root. For example, use this code to limit the search for 'FISH' to terms within the 'Fish Trait' branch of the FISHTRAITS Thesaurus:

```
<input type='text' name='n' data-bp_search_branch='Fish Trait' class='bp_form_complete-FISHTRAITS-name' />
```

Object types:

- **Search 'class' and 'property' objects.** Use `data-bp_objecttypes='class,property'`. For example, use this code to search for 'function' among 'class' and 'property' objects from the "Adverse Event Reporting Ontology":
 - ```
<input type='text' name='o' data-bp_objecttypes='class,property' class='bp_form_complete-AERO-uri' />
```
- **Search only 'property' objects.** Use `data-bp_objecttypes='property'`. For example, use this code to search for 'function' among only 'property' objects from the "Adverse Event Reporting Ontology":
  - ```
<input type='text' name='p' data-bp_objecttypes='property' class='bp_form_complete-AERO-uri' />
```
- **Search 'class' and 'individual' objects.** Use `data-bp_objecttypes='class,individual'`. For example, use this code to search for 'seizure' among 'class' and 'individual' objects from the "Adverse Event Reporting Ontology":
 - ```
<input type='text' name='q' data-bp_objecttypes='class,individual' class='bp_form_complete-AERO-uri' />
```
- **Search only 'individual' objects.** Use `data-bp_objecttypes='individual'`. For example, use this code to search for 'seizure' only among 'individual' objects from the "Adverse Event Reporting Ontology":
  - ```
<input type='text' name='r' data-bp_objecttypes='individual' />
```

```
class='bp_form_complete-AERO-uri'/>
```

Semantic artefact search widget

Developers can add a search box to their website that will allow users to search a specific EcoPortal semantic artefact. When users select the term of interest (with the help of the auto-complete feature), they can move directly to the EcoPortal summary page of the corresponding concept.

To add a search widget to a HTML page that allows users to search for a specific semantic artefact:

- download the [quick_jump.js](#) file and put it on your server;
- copy the code below and paste it to your HTML page

```
<div id='bp_quick_jump'></div>
<script type='text/javascript'>
  var BP_ontology_id = 'FISHTRAITS';
</div>
<script src='quick_jump.js' type='text/javascript' charset='utf-8'></script>
```

Note: To use Quick Jump across multiple semantic artefacts:

- enter a comma-separated list of semantic artefact ids: `var BP_ontology_id = 'FISHTRAITS, ENVTHES'`, or
- set the variable to 'all' to search all semantic artefacts in EcoPortal: `var BP_ontology_id = 'all'`.

Update the copied code by changing the path to the `quick_jump.js` file that has to point to the server location where you put the HTML file. For example, if you put the **quick_jump.js** file in the same directory as your HTML file, this is the code that you have to use:

```
<script type='text/javascript'>
  var BP_ontology_id = 'FISHTRAITS';
</div>
<script src='quick_jump.js' type='text/javascript' charset='utf-8'></script>
```

to limit the term search to a particular branch of a semantic artefact, include the following JavaScript in your page (remember to replace the conceptid with the ID for the term that

you want to use as the root of your branch): `var BP_search_branch = '{class_id}';`

To display definitions with the list of terms that are returned, include the following JavaScript in your page: `var BP_include_definitions = true;`

Semantic artefact visualisation widget

Developers can put a widget on their website that shows semantic artefact of interest, or a part of it, and obtain the same result as in the ["Visualize" tab in EcoPortal](#).

To add a widget visualising your semantic artefact in your HTML page, copy the code from the Widget page of your semantic artefact of interest. Here is an example:

```
<iframe  
src='http://ecoportal.lifewatchitaly.eu/widgets/visualization/?ontology=FISHTRAITS&class=SELECT_CLA  
SS&apikey=YOUR_API_KEY' frameborder='0'>  
</iframe>
```

EcoPortal Ontology Tree Widget

Developers can put a widget on their website that displays the semantic artefact tree, or a part of it, with the same result obtained in the "Visualize" tab in EcoPortal.

Description

The semantic artefact tree widget is an embeddable Flex application that allows users to interact with a tree to browse semantic artefacts. The widget can be configured to display a list of all the EcoPortal semantic artefacts, allowing users to select which one to view, or can be set to view a particular semantic artefact. Moreover, there are several JavaScript helper functions for the widget, allowing it to be used as a fully functional component in other applications.

Location

To use the tree widget, you have to add the following files:

- <https://ecoportal.lifewatch.eu/widgets/jquery.ncbo.tree.css>
- <https://ecoportal.lifewatch.eu/widgets/jquery.ncbo.tree-2.0.2.js>

or the minified versions:

- <https://ecoportal.lifewatch.eu/widgets/minified/jquery.ncbo.tree.min.css>
- <https://ecoportal.lifewatch.eu/widgets/minified/jquery.ncbo.tree-2.0.2.min.js>

Setup

jQuery is required in order to use the tree widget. If jQuery is not still in your project, you have to include it as follows:

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
```

Then, you have to create a div element to store the tree:

```
<div id='tree'></div>
```

initialise the tree with the following JavaScript code:

```
var tree = $('#tree').NCBOTree({apikey: 'YOUR_API_KEY',ontology: 'FISHTRAITS'});
```

The div you created will contain a tree showing the roots of the FISHTRAITS semantic artefact. Additional setup options include:

- **autoclose**: to close other branches of the tree when expanding a new one;
- **treeClass**: the class to identify the tree;
- **autocompleteClass**: the class to identify the autocomplete (search) input;
- **width**: the width of the tree;
- **startingClass**: to start with this class selected (and expanded to) in the tree;
- **startingRoot**: to start with this class as the only root displayed in the tree. This allows you to display a sub-tree or branch of your choosing;
- **ncboAPIURL**: to use a different NCBO REST instance (not normally used);
- **ncboUIURL**: to use a different NCBO UI instance (not normally used).

Callbacks and event triggers

Developers can interact programmatically with the tree by using JavaScript and jQuery. Callbacks can be passed when setting up the tree:

```
var tree = $('#tree').NCBOTree({apikey: 'YOUR_API_KEY',ontology: 'FISHTRAITS', afterSelect:
```

```
function(event, classId, prefLabel, selectedNode){console.log(classId);}}
```

The available callbacks with their arguments are:

- **beforeExpand**, arg: expandedNode
- **afterExpand**, arg: expandedNode
- **afterExpandError**, arg: expandedNode
- **afterSelect**, arg: classId, prefLabel, selectedNode
- **afterJumpToClass**, arg: classId

It is possible to listen to these callbacks as jQuery events:

```
tree.on('afterSelect', function(event, classId, prefLabel, selectedNode){console.log(classId);})
```

The events with their arguments are:

- **beforeExpand**, arg: event, expandedNode
- **afterExpand**, arg: event, expandedNode
- **afterExpandError**, arg: event, expandedNode
- **afterSelect**, arg: event, classId, prefLabel, selectedNode
- **afterJumpToClass**, arg: event, classId

Public methods

There are a variety of methods to call on the object that is returned when calling **NCBOTree()**. They can be used to get information about the state of the tree or to interact with it programmatically.

```
var tree = $('#tree').data('NCBOTree');
```

OR

```
var tree = $('#tree')[0].NCBOTree;
```

OR

```
var tree = document.getElementById('tree').NCBOTree;
```

Once you have the tree instance, you can call the following methods:

- **tree.selectedClass()**: returns an object that represents the currently-selected class. The object includes attributes for ID, prefLabel, and URL. URL is the REST

location of the class, and performing an HTTP GET on that URL will provide a JSON representation of the class;

- **tree.selectClass(classId)**: activates the class in the tree with the matching URI (classId). This will only work if the class is already visible;
- **tree.jumpToClass(classId)**: replaces the current tree with a version that is expanded with a path from the root to the given class. This method is triggered when selecting a class from the search field. The provided class will be selected in the tree when the tree returns;
- **tree.changeOntology(ontologyAcronym)**: replaces the current tree with the roots from the given semantic artefact.

Acknowledgements

The release has been funded by the FAIR-IMPACT project, LifeWatch Italy and CNR PON-IR "LifeWatchPLUS" project.

6. Document Revision History

Revision	Reason for and description of change	Author	Date
1.	First issue	Lucia Vaira	12-02-2025