



Publications Office

Framework Contract No OP/10688

PMKI

DLV2.4 - User Manual – PMKI

Subject Deliverable

Version 2.0

Release Date 18/11/2019

Filename DLV-2.4 User Manual v.2.0.docx

Document Reference UserManual-PMKI

Document History

DOCUMENT HISTORY			
Version	Release Date	Author	Description
2.0	2020/05/12	Armando Stellato, Tiziano Lorenzetti (infeurope)	Updates following updates in the platform, still as a draft
1.0	2019/11/18	Armando Stellato, Tiziano Lorenzetti (infeurope)	First official release for PMKI 2018 as a website, now ported here as a PDF document

REVIEWS			
Version	Date	Reviewer	Position

Table of Contents

1	SUMMARY	6
2	USER MANUAL OVERVIEW	7
2.1	SECTIONS INDEX	7
2.1.1	<i>Administrator</i>	<i>7</i>
2.1.2	<i>Visitor User</i>	<i>7</i>
3	ADMINISTRATOR	8
3.1	FIRST ACCESS	8
3.1.1	<i>Administrator registration.....</i>	<i>8</i>
3.1.2	<i>System Configuration</i>	<i>8</i>
3.2	ADMINISTRATOR DASHBOARD	9
3.2.1	<i>Introduction.....</i>	<i>9</i>
3.2.2	<i>System Configuration</i>	<i>10</i>
3.2.3	<i>Project Management.....</i>	<i>12</i>
3.2.3.1	The Projects table	12
3.2.3.2	Project status.....	13
3.2.4	<i>Contributions Management</i>	<i>14</i>
3.2.4.1	Introduction.....	14
3.2.4.2	The Contributions table.....	15
3.2.4.3	Contributions approval.....	15
4	VISITOR USER	18
4.1	CONTRIBUTIONS SUBMISSION	18
4.1.1	<i>Introduction.....</i>	<i>18</i>
4.1.2	<i>Metadata Contribution</i>	<i>19</i>
4.1.3	<i>Stable resource contribution</i>	<i>20</i>
4.1.4	<i>Development resource contribution</i>	<i>22</i>

4.2	DATASETS VIEW.....	23
4.2.1	<i>Data view</i>	24
4.2.1.1	Introduction.....	24
4.2.1.2	The Data Structure View.....	25
4.2.1.3	The Resource View	25
4.2.1.4	Search.....	28
4.2.1.5	Graph View	30
4.2.2	<i>Sparql editor</i>	34
4.2.2.1	Introduction.....	34
4.2.2.2	SPARQL query editor	34
4.2.2.2.1	Query editing.....	34
4.2.2.2.2	Results Area	35
4.2.2.2.3	Exporting results	35
4.3	SEARCH.....	36
4.3.1	<i>Introduction</i>	36
4.3.2	<i>Search results</i>	36
4.3.3	<i>Search filters</i>	37
4.4	ALIGNMENTS.....	38
4.4.1	<i>Introduction</i>	38
4.4.2	<i>Table view</i>	39
4.4.3	<i>Graph view</i>	41

1 SUMMARY

This document provides all the information related to the operations that can be performed on PMKI from the point of view of the final user.

A copy of this document in the form of web pages is pending clearance from the Publications Office and the ISA² program for being published on a dedicated community site (as per the twin project VocBench).

2 USER MANUAL OVERVIEW

The following details different aspects of the user experience on the PMKI portal.

2.1 SECTIONS INDEX

2.1.1 ADMINISTRATOR

Administrators have capabilities spanning from system configuration to projects and contributions management. The administrator is the only user that can be authenticated in the portal.

- [First Access](#)
- [Administrator dashboard](#)
 - [Project Management](#)
 - [Contributions Management](#)

2.1.2 VISITOR USER

The visitor user can explore the public projects on the PMKI portal and might contribute resources to the portal

- [Contributions submission](#)
- [The Datasets View](#)
 - [Data view](#)
 - [Sparql editor](#)
- [Search](#)
- [Alignments](#)

3 ADMINISTRATOR

3.1 FIRST ACCESS

3.1.1 ADMINISTRATOR REGISTRATION

PMKI requires that one (and only one) user is registered that is the system administrator. Once accessed for the first time, the system recognizes that no user has been registered, so it prompts a form in order to register the administrator user.

PMKI

PMKI has detected that no user has yet registered. In order to use the platform, you need to create an administrator account by entering personal data and login credentials.

Email	<input type="text" value="Email"/>
Password	<input type="password" value="Password"/> 👁
Confirm Password	<input type="password" value="Confirm Password"/> 👁
Given name	<input type="text" value="Given name"/>
Family name	<input type="text" value="Family name"/>

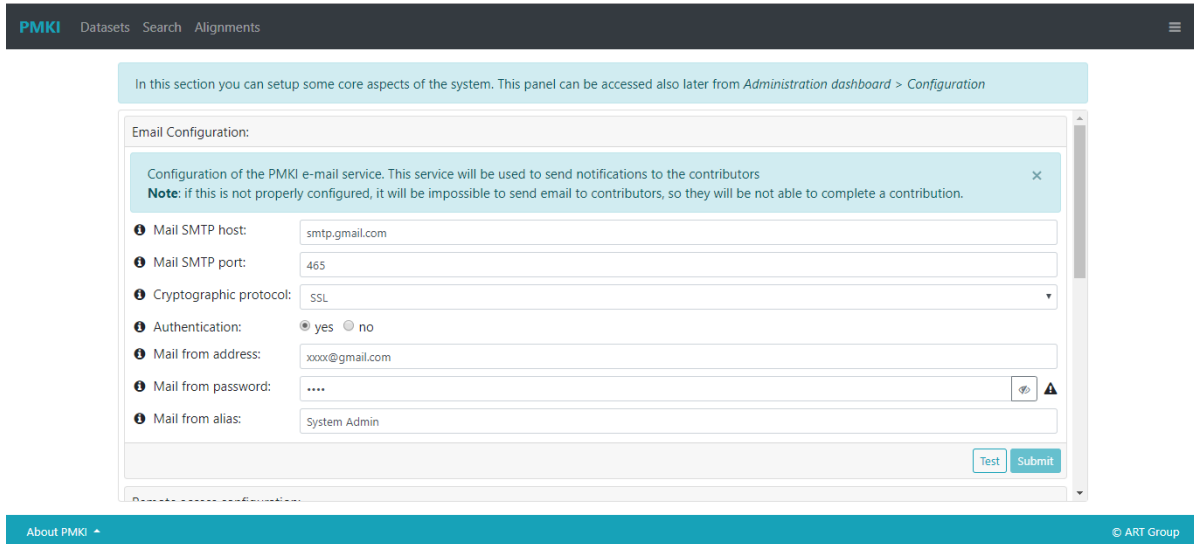
Submit

About PMKI ↗
© ART Group

3.1.2 SYSTEM CONFIGURATION

At the first access PMKI needs also to be configured in some core aspects of the system. These configurations are not mandatory but they are needed in order to make the whole system works as expected. Once the administrator registration is complete, the system redirects to a configuration page where following are requested:

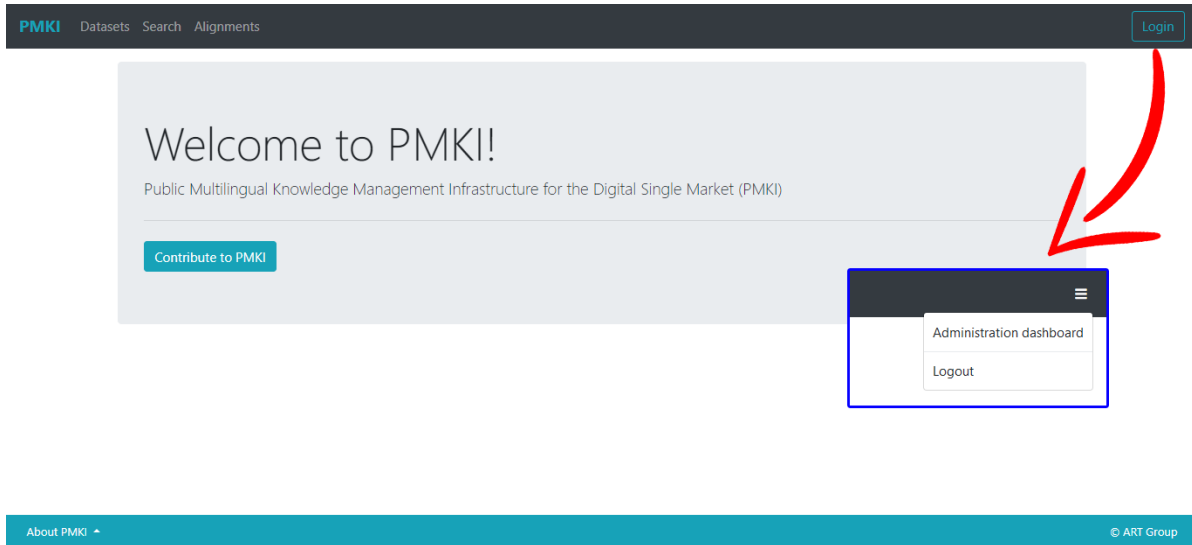
- *Email configuration*: configuration of the PMKI e-mail service: This service is used to send notifications to the contributors and to the administrator itself.
- *Remote access configuration*: configuration of the remote triple store that hosts the projects in PMKI (usually a GraphDB instance).
- *VocVench configuration*: Configuration of the VocBench/SemanticTurkey instance dedicated to the development of unstable resources.



3.2 ADMINISTRATOR DASHBOARD

3.2.1 INTRODUCTION

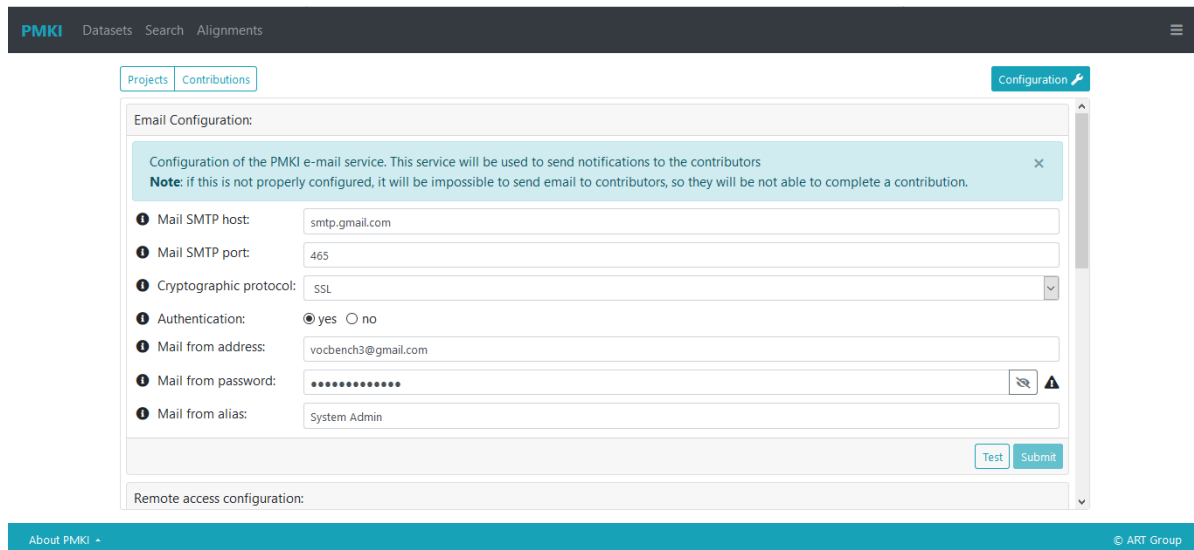
In PMKI the administrator is the only user authorized to manage some "behind the scene" aspects of the platform. The administrator user can login to the system through the *Login* button located on the top right corner of the application. Once the administrator is logged in, a dropdown menu takes the place of the *Login* button. From this menu it is possible to access the administration dashboard or to logout.



By clicking on the *Administration dashboard* item, the administrator landing directly on the *Projects manager* dashboard. Documentation about this dashboard is available [here](#). In this page, through the buttons on the top, it is possible to manage also the submitted [contributions requests](#) and to configure some core aspects of the system.

3.2.2 SYSTEM CONFIGURATION

From this panel it is possible to configure some core aspects of the system. These configurations are not mandatory but they are needed in order to make the whole system works as expected. The same configuration panel is prompted at the [first access](#) to PMKI.



The screenshot shows the PMKI Configuration interface. At the top, there are navigation tabs for 'Projects' and 'Contributions', and a 'Configuration' button with a pencil icon. The main content area is titled 'Email Configuration:' and contains a light blue informational box with a close button (X) that reads: 'Configuration of the PMKI e-mail service. This service will be used to send notifications to the contributors. Note: if this is not properly configured, it will be impossible to send email to contributors, so they will be not able to complete a contribution.'

Below the note, there are several configuration fields:

- Mail SMTP host:** smtp.gmail.com
- Mail SMTP port:** 465
- Cryptographic protocol:** ssl (dropdown menu)
- Authentication:** yes (selected radio button) / no
- Mail from address:** vocbench3@gmail.com
- Mail from password:** [masked with dots] (password field with eye icon and warning triangle)
- Mail from alias:** System Admin

At the bottom right of the configuration area, there are 'Test' and 'Submit' buttons. Below the email configuration, there is a section for 'Remote access configuration:' which is currently empty.

The footer of the page contains 'About PMKI' on the left and '© ART Group' on the right.

Three different configuration sections are foreseen:

Email configuration: configuration of the PMKI e-mail service: This service is used to send notifications to the contributors and to the administrator itself. This configuration is particularly important as if it is not properly configured, the system will not be able to send email notification to the contributors with the details required to complete the contribution.

- **Mail SMTP host:** the IP/URL of the SMTP mail server;
- **Mail SMTP port:** the port of the SMTP mail server;
- **Cryptographic protocol:** the cryptographic protocol used: could be *SSL*, *TLS* or *None* if not used;
- **Authentication:** specifies if the authentication is required in the mail server;
- **Mail from address:** the email address used to send administration notifications;
- **Mail from password:** the password of the previous email address (required only if *Authentication* is *Yes*);
- **Mail from alias:** the alias name shown in the sent emails.

Remote access configuration: configuration of the remote triple store that hosts the projects in PMKI (usually a GraphDB instance). If this is not properly configured, the PMKI portal will not be able to create remote projects for hosting stable datasets but it will be possible to create just local projects.

- **Server URL:** URL/IP of the triple store;
- **Username:** Username for authentication (if any);
- **Password:** Password for authentication (if any).

VocBench configuration: Configuration of the VocBench/SemanticTurkey instance dedicated to the development of unstable resources. If this is not properly configured,

the PMKI portal will not be able to connect with VocBench, so it will be impossible to accept contribution for under-development resources.

- *VocBench URL*: URL of the VocBench instance accessible by the contributors for the editing of development resources. This URL will be sent to the contributor in the email notification about accepted contributions;
- *SemanticTurkey Host*: Complete URL (protocol+hostname+port) of the SemanticTurkey host;
- *VocBench/SemanticTurkey admin email*: email address of the administrator user in VocBench;
- *VocBench/SemanticTurkey admin password*: password of the administrator user in VocBench.

For the email and VocBench configurations it is available a *Test* button useful to check that the provided info are valid. In the first case a sample email is sent to an address provided by the user; in the second, a login request with the given credentials is performed to the given SemanticTurkey host.

3.2.3 PROJECT MANAGEMENT

3.2.3.1 The Projects table

When the administrator access the *Administration dashboard* he is redirect directly to the *Projects manager* panel. This panel simply contains a table with the list of projects hosted in PMKI. Here follows a description of the table headers:

- *Project Name*: the name of the project, if clicked the administrator is redirected to the [Data](#) page of the project (only if the status is *Public* or *Staging*);
- *Open/Close*: tells whether a project has been opened inside the system. A project can be closed in order to release its resources. However, when a project is closed, it is not possible to access it (it should be still possible to include its resources in the [search](#) results, accordingly to the filters);
- *Model*: the type of the dataset, it can be a *SKOS* thesaurus or an *OntoLex* lexicon;
- *Lexicalization Model*: the type of lexicalization adopted, it can be *RDFS*, *SKOS*, *SKOS-XL* or *OntoLex*;
- *Location*: the location of the repository. It can be *remote*, if hosted on an external triple store, or *local* if hosted directly on the RDF4J repository of the underlying [SemanticTurkey](#) server. In case the location is remote, further information are shown as tooltip on a *i* icon;

- *Status*: the status of the project, it can be *Public*, *Staging*, or *Pristine*.

Project Name	Open/Close	Model	Lexicalization Model	Location	Status
Eurovoc		SKOS	SKOSXL	remote ⓘ	Public
Gemet		SKOS	SKOS	local	Pristine
Teseo		SKOS	SKOSXL	remote ⓘ	Public
Wordnet		OntoLex	OntoLex	remote ⓘ	Staging

3.2.3.2 Project status

As seen in the previous section, the last column of the table shows the current status of the projects. What exactly means the status? In order to answer to this question it is necessary to briefly explain how a project is created in PMKI.

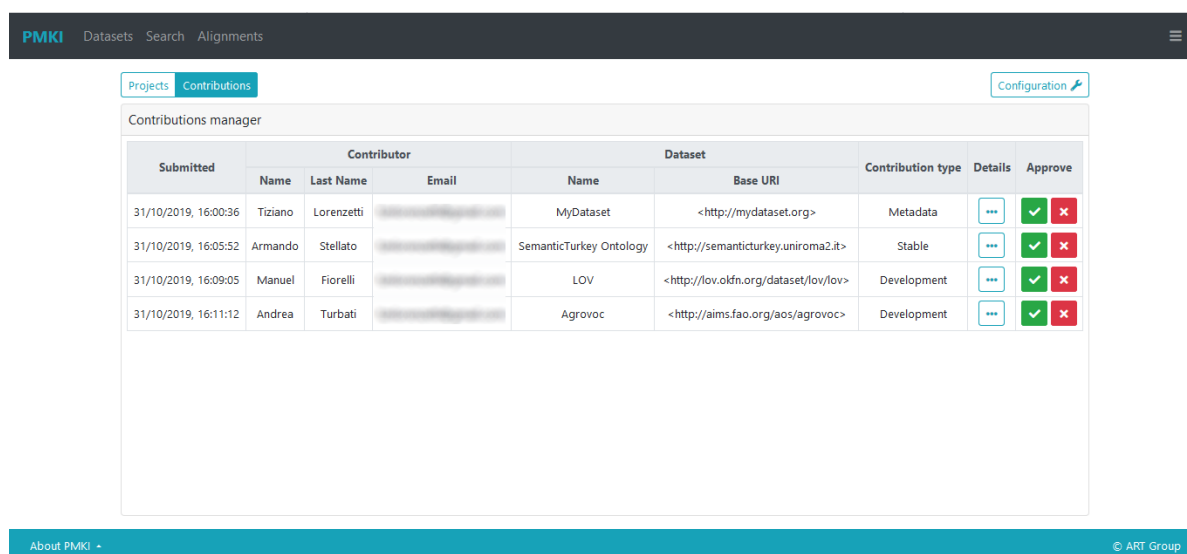
Currently in order to create a new project in PMKI is necessary to submit a contribution request for a [stable resource](#). Once the contribution request is evaluated and accepted by the administrator, a new project is created. This new created and empty project is then in the *Pristine* status. The contributor can then proceed to load the data into the project. Once the data is successfully loaded, the project is no more empty and moves from the *Pristine* status to the *Staging* one. A project in the *Staging* status is not visible to a "simple" user, a visitor, this means that in the [Datasets](#) page it will not appear in the datasets list and also the contained resources will not be returned by the [Search](#) feature. In short words, the *Staging* status can be described as private project, only visible to the administrator. In order to make visible the project with its content, the administrator needs to switch the status to *Public*.

A *Public* project can be later moved back again to the *Staging* status in order to restrict its visibility, and in the same way a *Staging* project can be made *Public* again. The only status that cannot be set or changed manually is the *Pristine* one, which is automatically assigned by the system once the project is created and then it's removed (in favour of the *Staging*) once the data is loaded by the contributor. The intermediate stage *Staging*, between *Pristine* and *Public*, has been introduced not just for handle the project visibility, but also for allowing the administrator to inspect the data that a contributor has loaded before to make it public and to eventually prevent incorrect data from being published on PMKI.

3.2.4 CONTRIBUTIONS MANAGEMENT

3.2.4.1 Introduction

When a user submits a contribution request an email notification is sent to the administrator. Then the administrator can examine the submitted requests in a dedicated page: the *Contributions manager*.



Submitted	Contributor			Dataset		Contribution type	Details	Approve
	Name	Last Name	Email	Name	Base URI			
31/10/2019, 16:00:36	Tiziano	Lorenzetti	[redacted]	MyDataset	<http://mydataset.org>	Metadata	...	✓ ✗
31/10/2019, 16:05:52	Armando	Stellato	[redacted]	SemanticTurkey Ontology	<http://semanticturkey.uniroma2.it>	Stable	...	✓ ✗
31/10/2019, 16:09:05	Manuel	Fiorelli	[redacted]	LOV	<http://lov.ckfn.org/dataset/lov/lov>	Development	...	✓ ✗
31/10/2019, 16:11:12	Andrea	Turbati	[redacted]	Agrovoc	<http://aims.fao.org/aos/agrovoc>	Development	...	✓ ✗

3.2.4.2 The Contributions table

A table lists all the submitted requests waiting to be evaluated. For each request are shown: the data of submission, basic information about the contributor, the name and the base URI of the dataset involved in the contribution and the contribution type.

The last two columns of the table are the more significant. The button under the *Details* column opens a modal dialog that allows to inspect all the information provided by the contributor during the request submission. In the figure below an example of details about a stable resource contribution.

The screenshot shows a modal dialog titled "Details" with a close button in the top right corner. The dialog is divided into two main sections: "Contributor" and "Contribution".

Contributor section:

- Name: Tiziano
- Last name: Lorenzetti
- Email: tizianorenzetti@gmail.com
- Organization: ART @Tor Vergata

Contribution section:

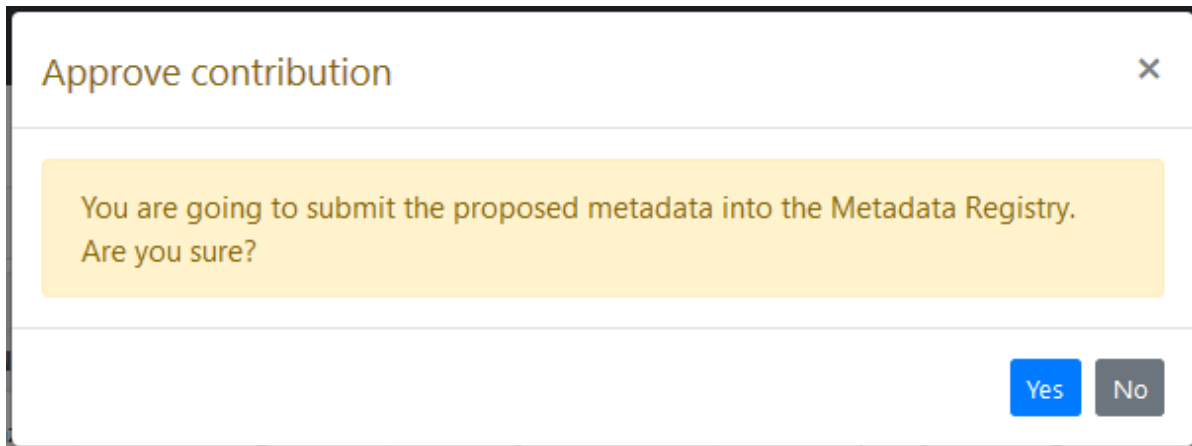
- Resource name: MyDataset
- Base URI: <http://mydataset.org>
- Identity: <http://mydataset.org#void.ttl>
- Dereferenciation system: Unknown
- Sparql endpoint: <http://mydataset.org/sparql> No aggregation
- URI space: http://mydataset.org/

A "Close" button is located at the bottom right of the dialog.

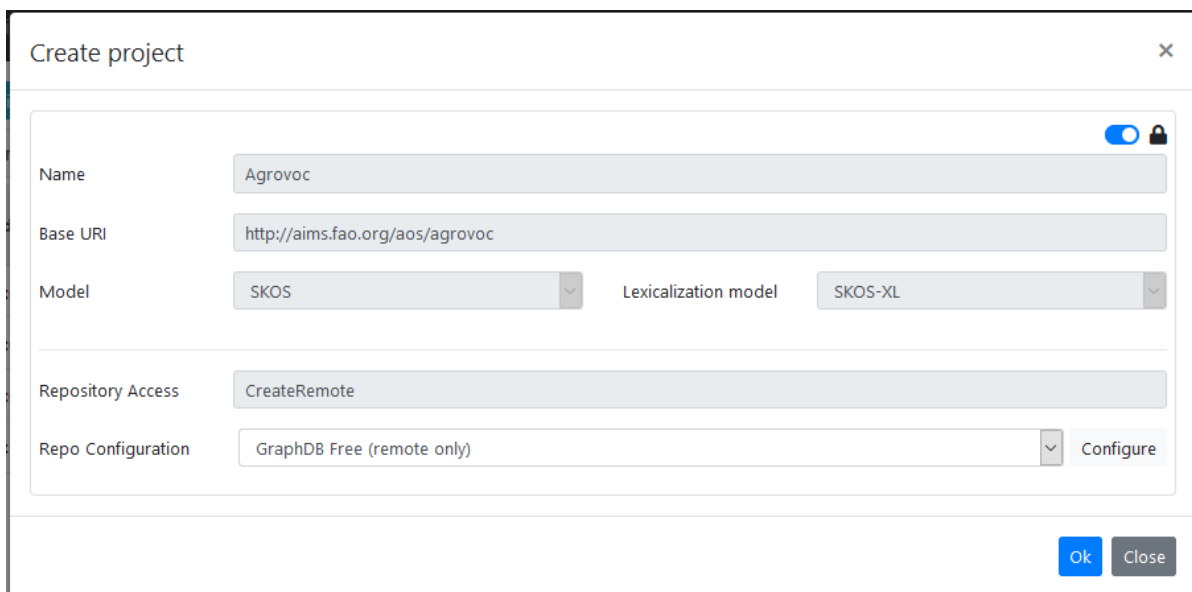
3.2.4.3 Contributions approval

The column *Approve* provide two buttons: the green one allows to accept the contribution, while the red one to reject it. The rejection of a request causes the removal of the same from the table and contextually an email is sent to the contributor in order to inform about the outcome of the evaluation. What happen when a request is accepted depends from the contribution type.

When a [metadata](#) contribution is accepted a simple dialog asks for confirmation to the administrator. In case the administrator accepts, the metadata proposed in the contribution are then written into the system Metadata Registry.



When a resource contribution is approved, be it a about [stable](#) resource or about one to [develop](#), a new project must be created.



A modal dialog prompt a project creation form with the following fields:

- **Name:** the name of the project. Any name which can be stored as a folder in the file system is a valid project name;
- **Base URI:** any valid ontology URI is accepted. If the uri ends with no trailing # nor /, the default namespace will end with a #. If the URI ends with /, the default namespace will be the same as the baseuri;

- *Model*: the data model adopted in the project. It can be SKOS (for thesauri) or OntoLex (for lexicons);
- *Lexicalization model*: it can be RDFS, SKOS, SKOS-XL or OntoLex depending on the lexicalization model you want to adopt for your data;
- *Repository Access*: one of:
 - *CreateLocal*: creates a local data repository;
 - *CreateRemote*: creates a repository on a remote triple store;
- *Repo Configuration*: configuration of the repository. Currently there are configurable settings for RDF4J and GraphDB stores.

The fields *Name*, *Base URI*, *Model* and *Lexicalization model* are pre-filled with the values provided by the contributor (the project name is eventually sanitized) and this values are readonly. The administrator anyway can enable the editing of these fields by unlocking them through the switch on the top right corner.

In case of development resource contribution, the selection of the *Repository Access* is forced to *CreateRemote* and cannot be changed. Recalling that in this case the project is created on a remote [configured](#) instance of VocBench, a *RemoteAccessConfiguration* needs to be [stored](#) on it (in case multiple configurations are found, the first will be used).

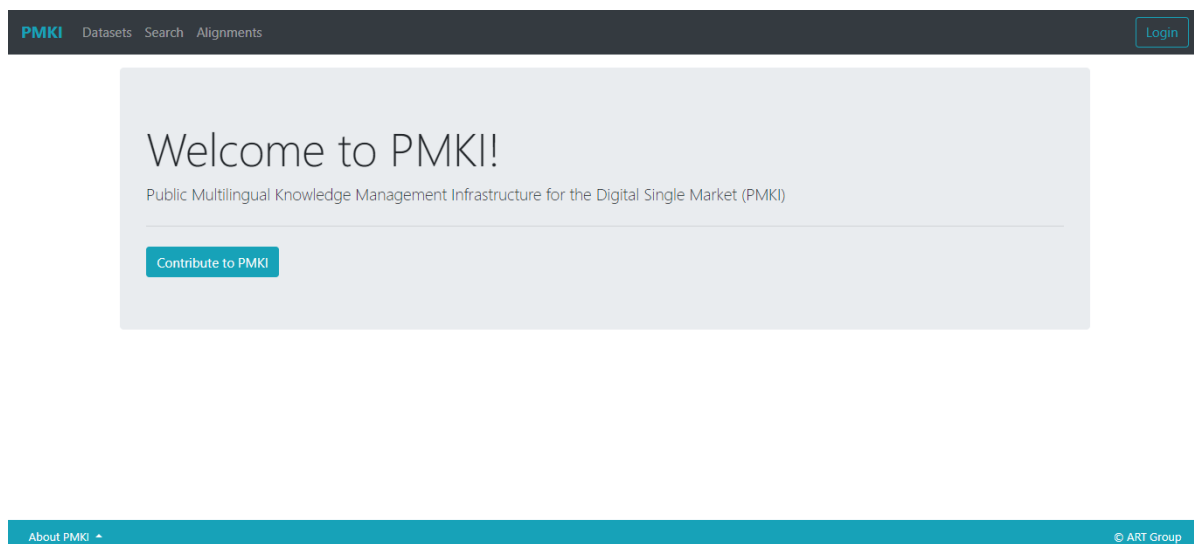
By clicking on the OK button, the new project is created and, if no error occurs during the procedure, an email notification is sent to the contributor containing details on how to proceed and complete the contribution process.

4 VISITOR USER

4.1 CONTRIBUTIONS SUBMISSION

4.1.1 INTRODUCTION

Any user can contribute to PMKI through the "Contribute to PMKI" button located in the home page.



In the contribution page a form asks for personal information of the contributor. The email address is quite important as it will be the same used for notifying the contributor about the outcome of the contribution evaluation by the administrator, if approved or rejected, and eventually for informing in how to proceed in order to complete the contribution.

Finally a *Contribution* field allow to choose the type of contribution that the user wants to request. There are three different types of contribution:

- *Metadata contribution*
- *Stable resource contribution*
- *Development resource contribution*

According to the selected one, the page shows a dedicated form in the blank space below. Once the form is properly filled, the contribution request can be submitted through the *Submit* button. Then the PMKI administrator will evaluate the request in order to approve or reject it.

For details about the contribution management from the administrator point of view, see the [dedicated page](#).

4.1.2 METADATA CONTRIBUTION

Metadata contribution allows the user to provide metadata about an existing resource available on the web.

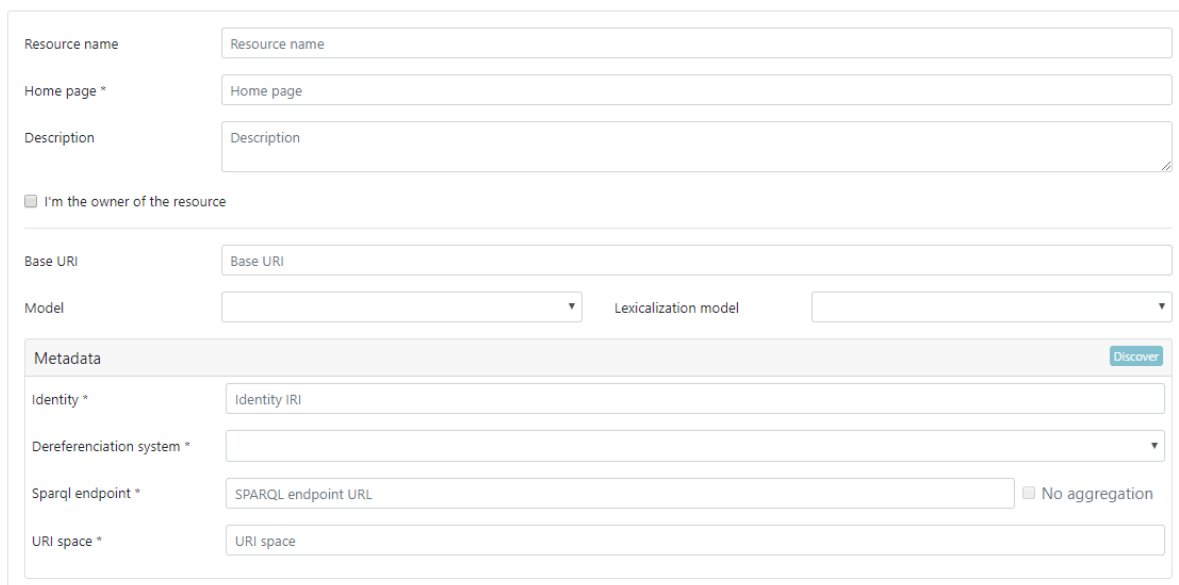
The following fields are requested by the submission form:

- *Base URI*: The Base URI of the dataset. It can be optionally used to exploit the automatic discovering of the metadata simply by clicking the *Discover* button nearby.
- *Resource name*: The name of the resource.
- *Identity*: An IRI that identifies the dataset (optional, if not provided an IRI is automatically generated by the system).
- *Dereferenciation system*: Tells if the IRI of the resources defined by the dataset can be dereferenced (optional, unspecified is equivalent to *unknown*).
- *Sparql endpoint*: IRI of the SPARQL endpoint, if available (optional).
- *URI space*: An URI that is a common string prefix of all entity URIs in the dataset. In other words, the dataset namespace (optional).

If the contribution is approved by the administrator, the metadata is directly written to the metadata registry of PMKI.

4.1.3 STABLE RESOURCE CONTRIBUTION

This contribution allows to contribute a mirrored version of a resource already published on its own portal.



The screenshot shows a web form for resource contribution. It includes the following fields and controls:

- Resource name**: Text input field.
- Home page ***: Text input field.
- Description**: Text area with a rich text editor interface.
- I'm the owner of the resource**: A checkbox.
- Base URI**: Text input field.
- Model**: Dropdown menu.
- Lexicalization model**: Dropdown menu.
- Metadata**: A section header with a **Discover** button.
- Identity ***: Text input field.
- Dereferenciation system ***: Dropdown menu.
- Sparql endpoint ***: Text input field with a **No aggregation** checkbox.
- URI space ***: Text input field.

The following generic fields can be provided in the form:

- *Resource name*: The name of the resource.
- *Home page*: The website home page of the resource, if available (optional).
- *Description*: A description of the contributed resource. This is useful in order to inform the administrator of the treated domain of the resource.

Eventually, the contributor can also inform the administrator that he is the owner of the resource by ticking the dedicated checkbox.



More technical fields are also available:

- *Base URI*: The Base URI of the dataset. It can be optionally used to exploit the automatic discovering of the metadata simply by clicking the *Discover* button nearby.
- *Model*: The data model adopted by the resource. Those available are *SKOS* and *Ontolex*.
- *Lexicalization model*: The lexicalization model adopted by the resource. Those available are *RDFS*, *SKOS*, *SKOS-XL* and *Ontolex*.

Finally, metadata can be also provided in the same way of the *Metadata contribution* described above.

If the request is approved by the administrator, a temporarily-private empty project (named after the *resource name* provided) is created waiting for the data to be loaded. The contributor will then receive an email with a link for loading the RDF data about the resource.

Load data

Email address 	<input type="text" value="Email address"/>	
Project name 	<input type="text" value="Project name"/>	
File	Browse <input type="button" value="Select a file..."/>	
Lifter	<input type="text" value="RDF Deserializing Lifter"/>	Format <input type="text" value="RDF/XML"/>
Resolve imports	<input type="text" value="From Web"/>	
		<input type="button" value="Load"/>

The form presented in the loading page asks for the contributor *Email address* and the *Project name* (written in the email) needed in order to cross checking them with the information provided during the contribution request submission. Here an RDF *File* needs to be provided as well by browsing the local filesystem. The selection of the file automatically focus the *Format* selector to the suggested choice. The *Resolve imports* combo box shows the following possible values, instructing the system on how to import vocabularies specified on transitive dependencies, that is, vocabularies that are owl:imported by the loaded data, or by other vocabularies in turn imported by it.

- *From Web*: transitive imports are resolved by trying to import the related vocabularies through their URIs specified in the owl:imports declarations.
- *From Web with fallback to Ontology Mirror*: if a vocabulary required by a transitive import is not found on the Web, then the Ontology Mirror will be used as a fallback solution.

- *From Ontology Mirror*: transitive vocabulary dependencies will be resolved locally on the Ontology Mirror.
- *From Ontology Mirror with fallback to Web*: if a vocabulary required by a transitive import is not found on the Ontology Mirror, then the Web will be used as a fallback solution

Once the data is successfully loaded, the resource is again subject of evaluation by the administrator that can decide if accept it, and make the dataset public, or reject it.

4.1.4 DEVELOPMENT RESOURCE CONTRIBUTION

This contribution allows to contribute a resource that is not already available as linked data. Eventually is available the conversion from non-RDF formats.

The screenshot shows a web form with the following fields:

- Conversion required**: A dropdown menu with a small square icon on the left and a downward arrow on the right.
- Name**: A text input field containing the placeholder text "Resource name".
- Home page ***: A text input field containing the placeholder text "Home page".
- Description**: A text input field containing the placeholder text "Description".
- Base URI**: A text input field containing the placeholder text "Base URI".
- Model**: A dropdown menu.
- Lexicalization model**: A dropdown menu.

Besides the same information provided in the *Stable resource contribution*, except for those regarding the metadata, there is a further optional field *Conversion required* that allows to choose among: *Excel*, *TBX* and *Zthes*.

If the request is approved by the administrator, an empty project (named after the *resource name* provided) is created on a configured instance of [VocBench](#) dedicated to the development of these resources. Then an email will be sent to the contributor with the instruction on how to proceed. The workflow now depends on the previously chosen *conversion required*:

- In case of *TBX*, *Zthes* or no conversion required at all, the email contains a link to the loading page and the name of the created project. The loading page is for the most the same of the one just seen for the stable resource contribution, it just differs for the *Lifter* chosen according to the conversion type (not changeable). Once the data is successfully loaded, another email is sent to the contributor with the details for accessing the VocBench instance, namely the link

and the credentials of a new created user (if there is no user already registered for the given contributor email).

- Otherwise, in case of *Excel* conversion required, the email contains the link to the VocBench instance and the credentials for accessing it (also in this case the VB user is contextually created if there is no user already registered for the given contributor email). From VocBench the contributor is then able to exploit the tool [SheetRDF](#) for transforming the spreadsheet.

In all these cases described, the user can edit the resource as desired on VocBench and then, when ready, download it and submit a new stable resource contribution request to PMKI.

4.2 DATASETS VIEW

The *Datasets* page presents the list of all the public datasets stored in the PMKI platform.

In the left side of the page it is possible to filter the datasets according their types (KOS or lexicons) and to filter out or show the closed projects. Closed projects, if any, are not explorable and are light gray colored in the list.

On the top of the datasets list a search bar allows to filter the datasets. The entered text is searched into the project names and into the base URIs, and only those projects that contain it are shown.

For each project in the list it is shown the name, colored in cyano if the project is open, in light-gray otherwise. By clicking the name the user is redirect to the [Data](#) page (only for open projects). Under the name there is the base URI in the bottom left corner. The top right corner of the project "card" indicates the data model and the lexicalization model adopted by the dataset. Finally, in the bottom right corner there is a button that launches the creation of an index required by the global [search](#).

PMKI Datasets Search Alignments

Browse PMKI datasets

Dataset type

- KOS
- Lexicons
- Only open projects

Search

Eurovoc http://eurovoc.europa.eu/	Model: SKOS Lexicalization: SKOSXL
Gemet http://www.eionet.europa.eu/gemet/	Model: SKOS Lexicalization: SKOS
Teseo http://www.senato.it/teseo/tes	Model: SKOS Lexicalization: SKOSXL
Wordnet http://art.uniroma2.it/pmki/omw/	Model: OntoLex Lexicalization: OntoLex

About PMKI | © ART Group

4.2.1 DATA VIEW

4.2.1.1 Introduction

The Data view page is accessed after clicking on a dataset name in the [Datasets](#) page. The Data page is composed of two main areas: the structure on the left, and the resource view on the right.

PMKI Datasets Search Alignments Login

Eurovoc

Data
Sparql

Concept Collection Scheme Property Alignments

- AAMS countries (en)
- ▼ ● access to the courts (en)
 - legal aid (en)
 - local access to the law (en)
 - right of action (en)
- ▶ ● accounting (en)
- ▶ ● accounting system (en)
- ▶ ● ACP countries (en)
- ▶ ● administrative law (en)
- ▶ ● administrative personnel (en)

Search...

● access to the courts (en) <http://eurovoc.europa.eu/5400>

Type	skos:Concept
Top Concept Of	1221 justice (en) EuroVoc (en)
In Scheme	1221 justice (en) EuroVoc (en)
Preferred Label	достъп до правосъдие (bg) přístup k soudu (cs) adgang til retsvæsenet (da) Zugang zur Rechtspflege (de) πρόσβαση στη δικαιοσύνη (el) access to the courts (en) acceso a la justicia (es) õiguskaitse kättesaadavus (et) oikeussuojakeinot (fi)


About PMKI | © ART Group

4.2.1.2 The Data Structure View


The left section of the data view page is dedicated to the data structures. This is a panel containing tabs with tree or list structures. The available tabs, that might differ according to the data model adopted in the dataset (SKOS or Ontolex), are:

- *Concept*: contains a panel showing the *skos:Concepts* of the dataset (hierarchically structured or as a plain list, based on the selected visualization mode), eventually filtered according to the selected active schemes.
- *Collection*: contains a panel with the *skos:Collections* of the dataset (available only in SKOS datasets).
- *Scheme*: contains a panel with the *skos:ConceptSchemes* of the dataset (available only in SKOS datasets).
- *Property*: contains a panel with the properties tree.
- *Lexicon*: contains a panel with the list of the *lime:Lexicons* (available only in Ontolex datasets).
- *LexEntry*: contains a panel with the *ontolex:LexicalEntry* in the active lexicon (available only for Ontolex datasets).
- *Alignments*: this is a "special" tab. It contains a list of aligned datasets with the current one. Further details can be found in the [dedicated page](#).

All structure areas in the Data View have two buttons in common (upper-right part of the structure area).

The Rendering Button  activates the rendering for the resources shown in the structure.

The Refresh Button  allows the user to refresh the view on the structure.

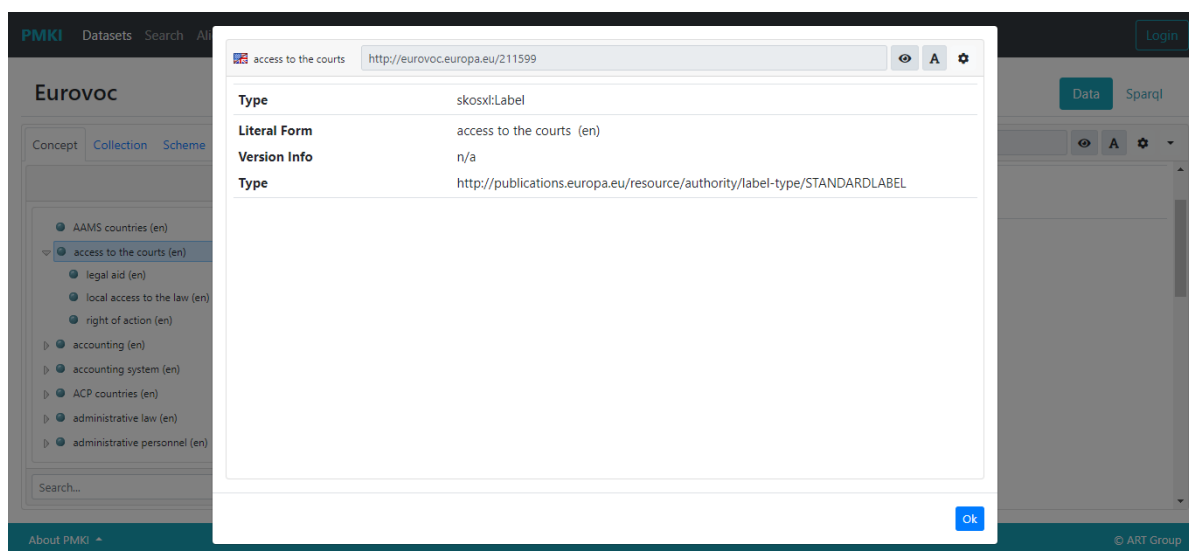
A further gear button  is present in the *Concept* and *LexEntry* panel and allows to edit the settings of the specific structure. Currently, the only one available is for the visualization mode of the resources.

At the bottom of every structure panel (so, except for *Alignments*) there is a search bar that will be described later.

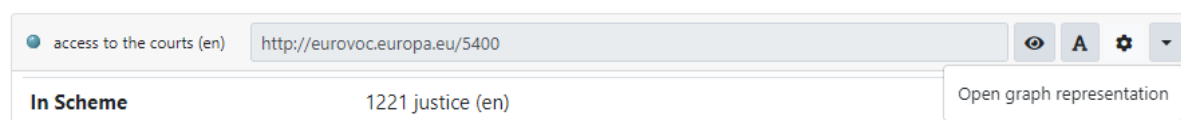
4.2.1.3 The Resource View

The ResourceView is a general widget for showing the details of different kind of resources. Everytime a resource is clicked in the left structure view, it is described in it. Even the resources that appears as values in a ResourceView can be in turn inspected

simply by double-clicking it over, contextually it is focused in the structure view. If the value resource is not reachable in any tree or list its ResourceView will be open in a modal dialog as in the figure below.



At the top of the ResourceView there is a bar with three buttons: the button with the eye allows to enable or disable the inferred information; the button with the capital **A** enables/disables the rendering; the one with the cog icon allows to edit the ResourceView settings. Next to these three buttons there is an arrow that expands a menu. From here it is possible to show the current resource in a graph view that will be described later.



In the ResourceView settings dialog it is possible to select the languages that will be shown when filtering literal values in resource descriptions. For instance, if the user wants to see just the english labels of a resource, it is enough to check only the *en* language and activate the filter through the switch in the top right corner.

ResourceView settings ✕

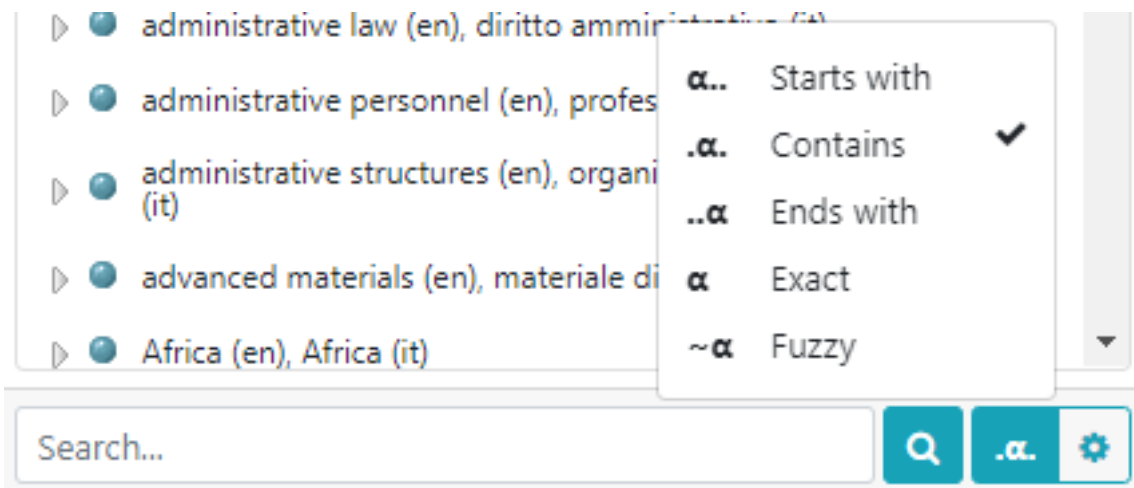
Value-filter Languages: ⚠

Select the languages that will be shown when filtering literal values in resource descriptions. Toggle the filter in order to hide the values with the unselected languages.

Active	ISO code <small>!?</small>	Language <small>↕</small>
<input type="checkbox"/>	ar	Arabic
<input type="checkbox"/>	bg	Bulgarian
<input type="checkbox"/>	bn	Bengali
<input type="checkbox"/>	bs	Bosnian
<input type="checkbox"/>	ca	Catalan
<input type="checkbox"/>	cs	Czech
<input type="checkbox"/>	da	Danish
<input type="checkbox"/>	de	German
<input type="checkbox"/>	el	Greek

4.2.1.4 Search

Navigation in tree and list panels is supported by a search bar for finding resources in the dataset.

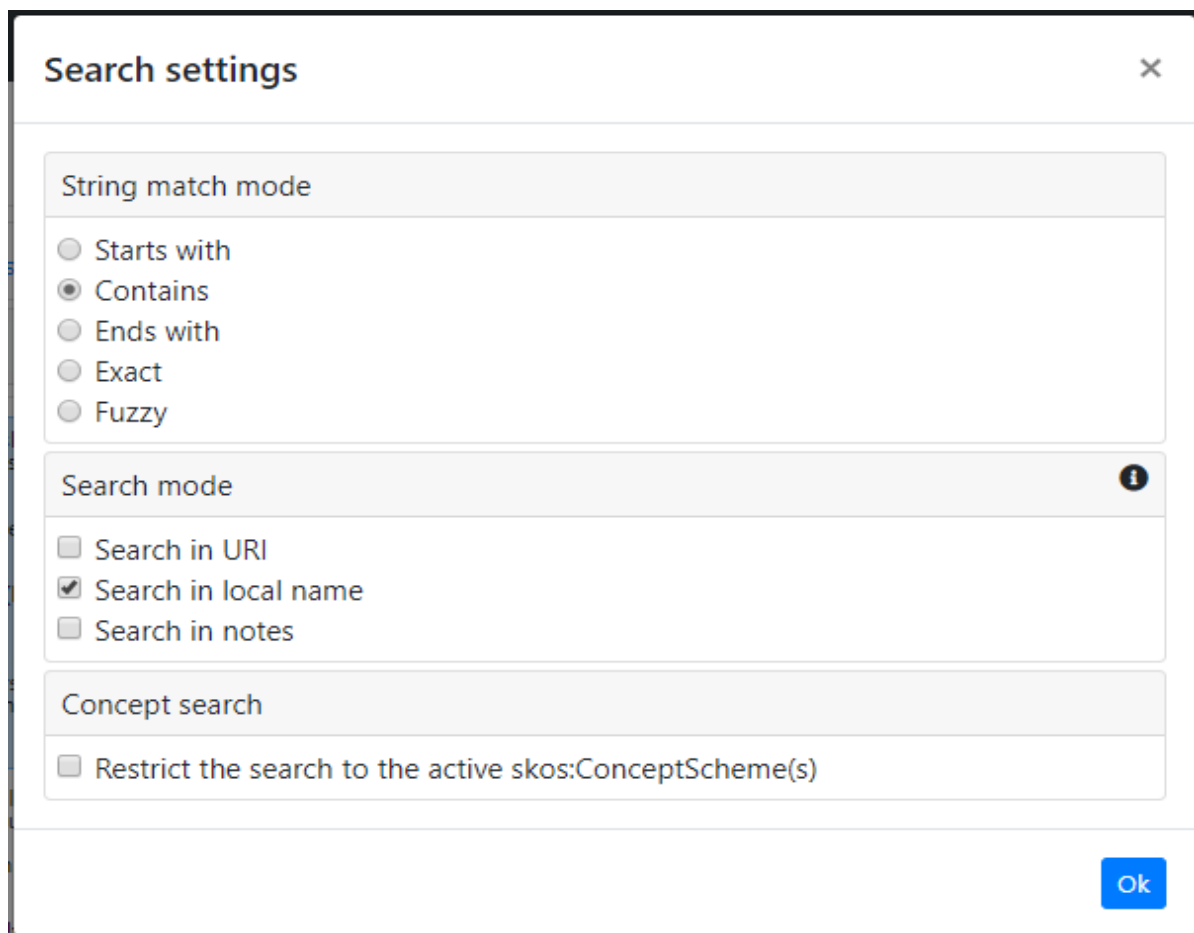


The search is fired when the "lens" symbol is clicked or the *ENTER* key is pressed during the typing into the search bar input field.

Besides the search button, a quick option button allows to choose the strategy for search, among:

- **Starts with:** looks for resource having at least a lexicalization starting with the prompted string
- **Contains:** looks for resource having at least a lexicalization containing the prompted string
- **Ends with:** looks for resource having at least a lexicalization ending with the prompted string
- **Exact:** looks for resource having an exact match with the prompted string
- **Fuzzy:** looks for resource having an approximate match with the prompted string

The gear button on the Search bar opens a further set of options:



The following options are available:

- The first option (String match mode) is related again to the search strategy, also available as a quick option on the bar.

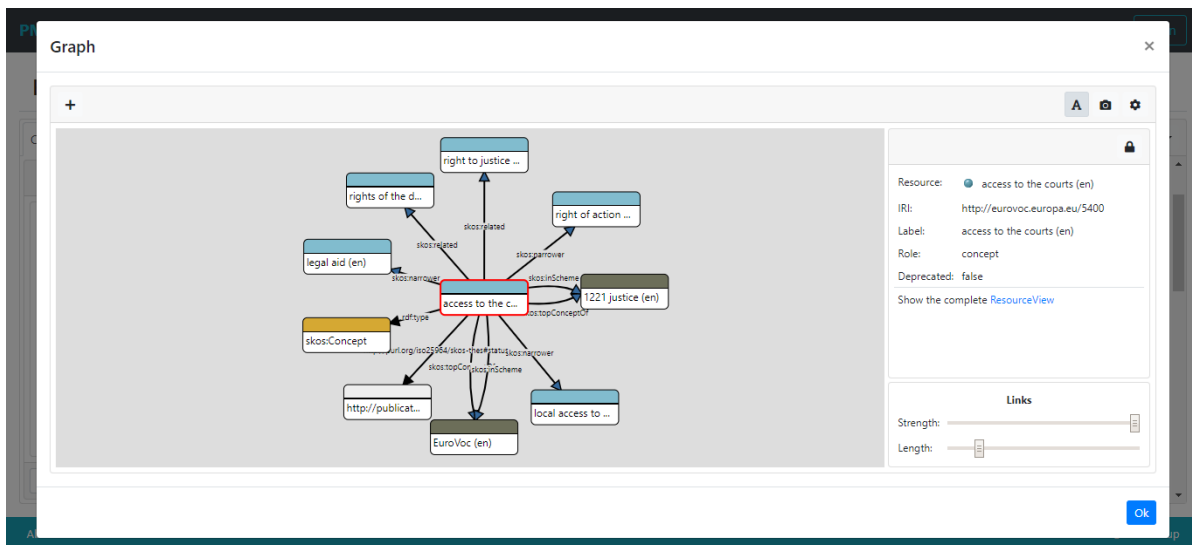
- `Search mode`: if the search (normally acting on lexicalizations only) should be extended to look into the whole URI, the local name of the resource and eventual notes.
- `Concept search` (available only in concept panel): through an option contained in this section it is possible to constrain the search to only the currently active `skos:ConceptScheme(s)`.

4.2.1.5 Graph View

In addition to the ResourceView, PMKI provides a further way to explore the data: the Graph View. This view allows the interactive visualization of the data with a force-directed graph (based on the [d3.js](#) library). The user can interact with the graph by selecting elements, exploring nodes and customizing the visualization.

In the graph the nodes represent resources and the edges represent properties. Basically, each node-edge-node connection represents a subject-predicate-object statement.

The Graph View can be accessed from the top bar of the ResourceView as seen before. The graph is rooted non the resource inspected in the ResourceView and it is expanded immediately.



The Graph view is basically composed by a single panel splitted in two main sections: the left one which shows the graph and the right one containing a panel that shows a brief description of the selected node. At the bottom of the right panel there are also two sliders that allow to change the strength and the length of the links.

Each node of the graph is represented by a rounded-corners rectangle with a top stripe. This stripe gives an immediate idea of the nature of the resource described in the node. It could be seen as an alternative to the resource icon used elsewhere, indeed, as shown by the previous figure, the selected node represents a concept and its top stripe has a light-blue color, very similar to the concept icon.

The edges represent the predicates whose arrow indicates the direction of the relation: from the subject to the object. As well as for the nodes, also the edges adopted a color code to explicit the nature of the predicate (as much as possible more similar to the color of the icons), for example the object properties are rendered with a blue arrow, the annotation properties with a yellow one, the datatype properties with a green one, and so on.

Every element of the graph, be it a node or an edge, when clicked is highlighted with a red border. Then the resource represented by the node is briefly described in the right panel from where it is possible to open the complete ResourceView. On the top on this panel, a lock button allows to lock/unlock the node position and so avoid that the node moves around the graph "pushed" by the graph forces. Every node can be expanded by double-clicking on it and collapsed in the same way.

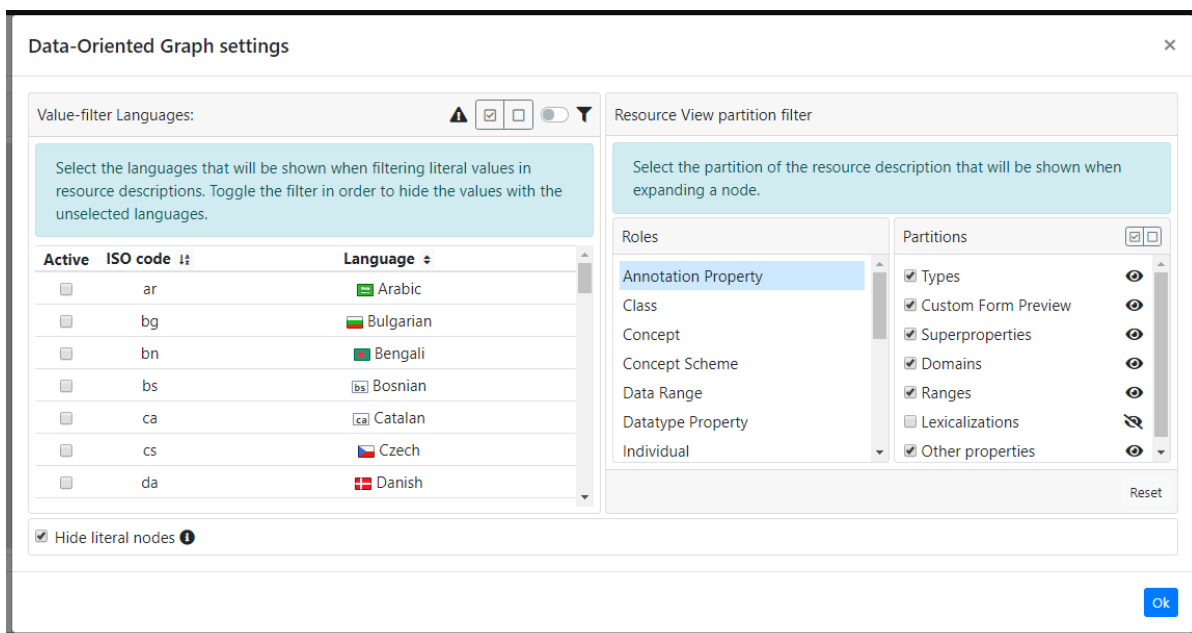
On the left side of the panel heading a "plus" button allows to drag-in a new node choosable by exploring the same tree/list which the root resource belongs to (e.g. the concept tree if the graph is rooted on a concept).

On the right side there are three other buttons:

- Rendering (the A button): activates/deactivates the rendering of the nodes and edges labels.
- Export (the "camera" button): exports a snapshot of the graph in an .svg file. Note that the exported SVG shows the graph exactly how it is visible in the panel, namely if any graphic element is cutted off from the gray area, it will be shown at the same way in the .svg file.
- Settings (the "gear" button): allows to edit the settings of the graph, in particular the filters on the visible elements.

With a growing amount of visual element (nodes and edges), the Graph View could be confused and it could reflect in a decrease of performance, so, in the attempt to provide a better user experience, in the data-oriented Graph View has been provided a filter mechanism. There are available two kind of filter: a global filter, that is applied whenever a user expands a node, and a local one, duly applied when the expansion of a node would produce too many nodes.

The "global" filters can be managed in the settings modal dialog.

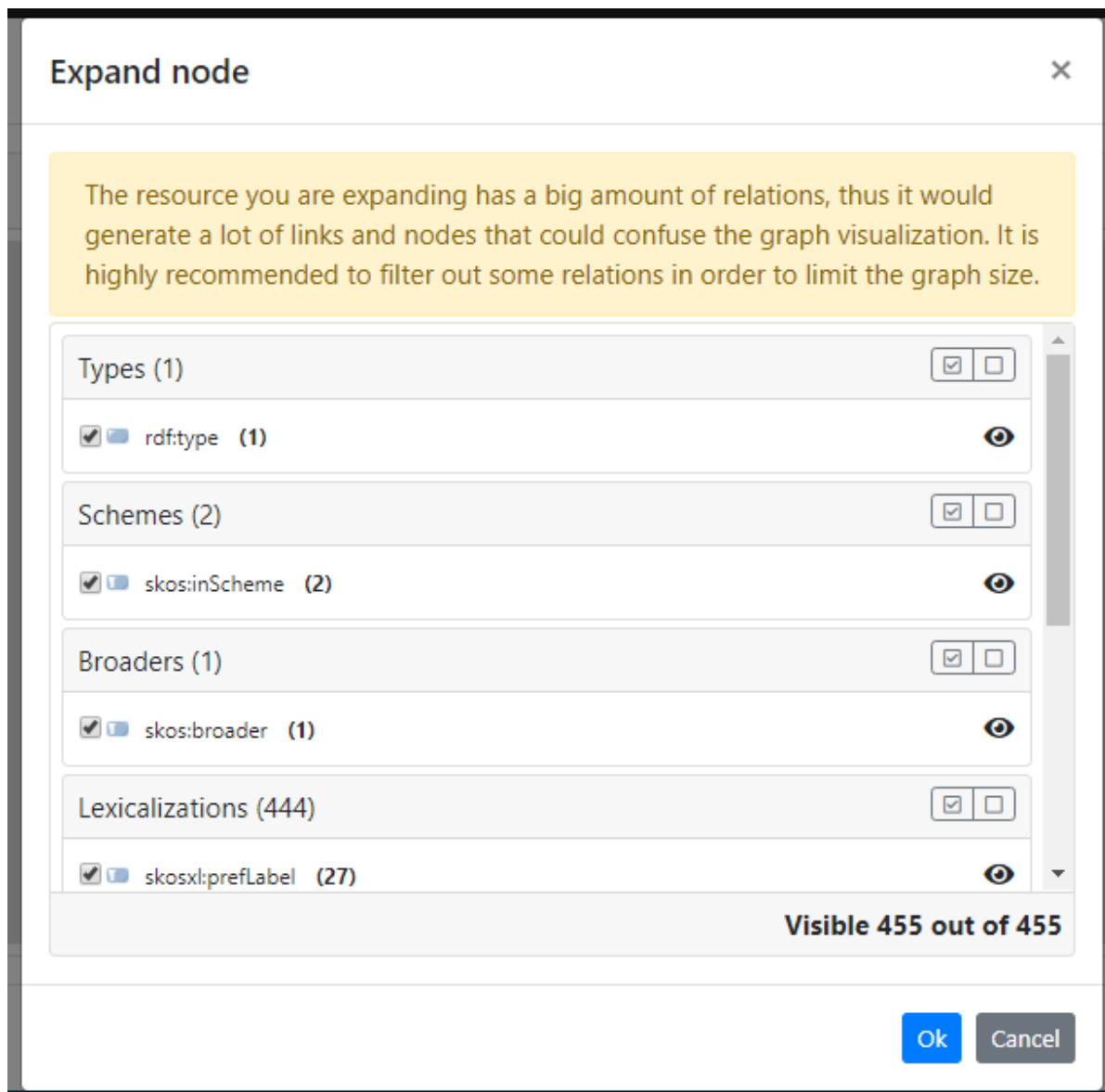


The panel *Value-filter Languages*, on the left side, allows to limit the literal nodes in the graph. Indeed, from the languages list it is possible to enable/disable the visualization of literal nodes (or generally language tagged resources) having the specified language, or even completely hide them all through the checkbox on the bottom. Note that this filter is the same used in the ResourceView, so changes applied here will affect the the ResourceView and in the same way changes to the ResourceView filter will affect the graph.

The right panel allows to customize the expansion of the nodes according the role of the described resource. For each role, it is possible to let expand or to cut off an entire

partition. For example, unchecking the *Lexicalizations*, all the relations that belongs to this partition (e.g. *rdfs:label*, *skos:prefLabel*, ...) will be excluded from the graph.

In some cases, the global filters might be not enough to limit the amount of element in the graph. If a resource has a large amount of relations, the expansion of the related node could produce a lot of links and nodes, so it will be proposed to filter the relations to be rendered. As shown in the figure below, a panel will let the user choose the partitions (or more specifically, the relations) to render.



4.2.2 SPARQL EDITOR

4.2.2.1 Introduction

The Sparql page can be accessed through *Sparql* button located on the top right corner of a dataset page.

This page is based on a tabbed panel that allows to open up to 11 sparql editor.

The screenshot shows the PMKI Sparql editor interface. At the top, there are navigation links for 'PMKI', 'Datasets', 'Search', and 'Alignments', along with a 'Login' button. The main header displays 'Eurovoc' and a 'Sparql' button. The query editor area contains a text area with the following SPARQL query:

```

1 PREFIX :↔
16
17 SELECT WHERE {
18   ?s ?p ?o .
19 } LIMIT 10
    
```

Below the query editor, there are buttons for 'Submit', 'Clear', and 'Include Inferred'. The results section shows a table with columns 's', 'p', and 'o' containing URIs. The results count is 10, and the query was executed in 156 milliseconds.

4.2.2.2 SPARQL query editor

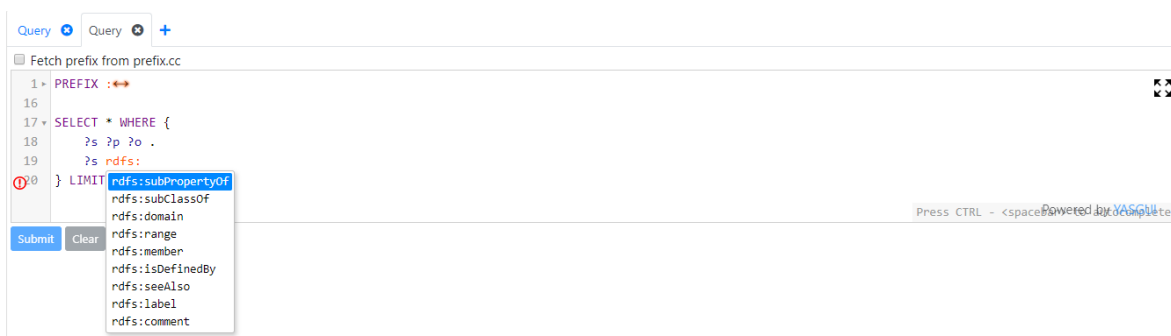
This editor is divided into three parts:

- the upper part provides the text area for query editing
- the central part contains the buttons for clearing/submitting the query, the option to include inferred statements in the result and exporter for saving the results of the queries in different formats.
- the bottom part reports the results of the query

4.2.2.2.1 Query editing

The query editing section of the SPARQL Editor contains a text area for typing the query. A syntax completion helper (see figure below, where RDFS predicates are being suggested) is automatically activated at each section of the query where it might provide suggestions. Syntax checking and highlighting are also features of this editor.

The syntax completion helper and syntax checking/highlighting features have been made available by adapting the open source [YASGUI](#) SPARQL editor to the VocBench environment. Besides the inclusion in the environment of PMKI, the completion has been improved to provide not only language syntax completion, but also (to a limited extent) live suggestions from the data present in the managed project (e.g. available prefix/namespace mappings).



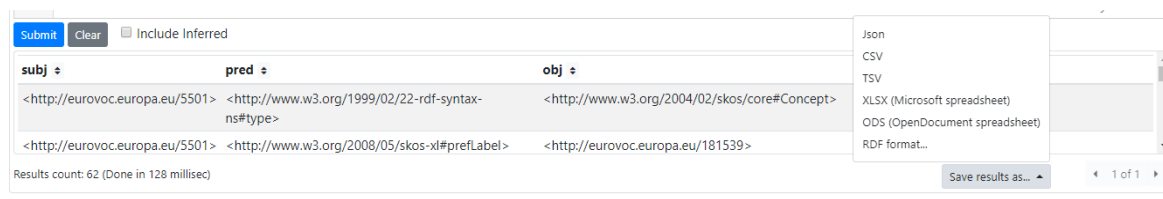
4.2.2.2 Results Area

The results area of the SPARQL Tab shows the results of the submitted query:

- `SELECT` queries provide results in the form of TUPLE bindings, so they are shown as records in a table, with the headers representing the bound variables
- `DESCRIBE` and `CONSTRUCT` queries are said to be `GRAPH` queries, in that the result of their processing is a RDF GRAPH. For this reason, a table with a static template (three headers representing the three elements of a RDF triple) is used to represent results content
- `ASK` queries provide a *boolean* result, telling if the `GRAPH PATTERN` of the query successfully matched the ontology graph

4.2.2.3 Exporting results

The results obtained from a SPARQL query can be exported to an external file.



Currently the following formats are supported:

- *json*
- *text* (CSV and TSV)
- *spreadsheet* (XLSX Microsoft format and ODS Open Document Spreadsheet format)
- *RDF* (only for graph queries, such as DESCRIBE and CONSTRUCT)

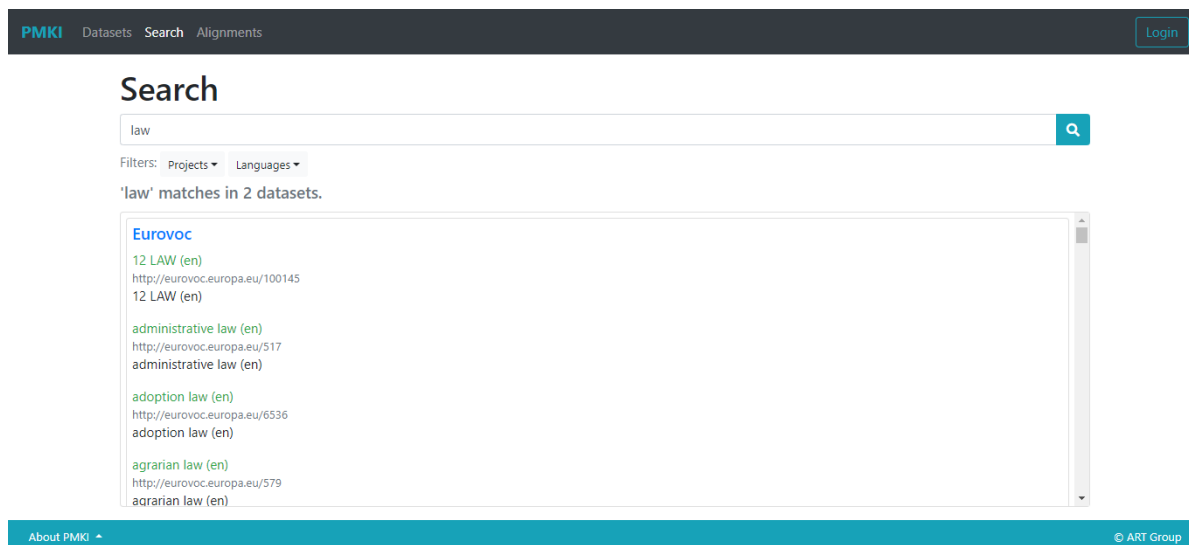
4.3 SEARCH

4.3.1 INTRODUCTION

The *Search* page can be accessed from the top navigation bar. This page allows to search resources that match a given string among all the dataset in PMKI.

4.3.2 SEARCH RESULTS

The search page is a clean page with just a search bar.



The search is fired when the "lens" symbol is clicked or the *ENTER* key is pressed during the typing into the search bar input field.


The results are grouped by dataset which the name, in blue, if clicked redirects to the related [Data](#) page.

Each results shows:

- the rendering of the matched resource, in green, which if clicked redirects to the Data page where the resource is focused in the tree/list structure showing the ResourceView;
- the IRI of the resource, in light-gray, which is clicked behaves the same of the previous;
- the value that matched the search string: it could be a label or a note of the resource.

4.3.3 SEARCH FILTERS

Just between the search bar and the space dedicated to the results, there is a stripes where it is possible to customize the search filters.

Search 

Filters: Projects Languages

Any language

de

en

es

fr

it

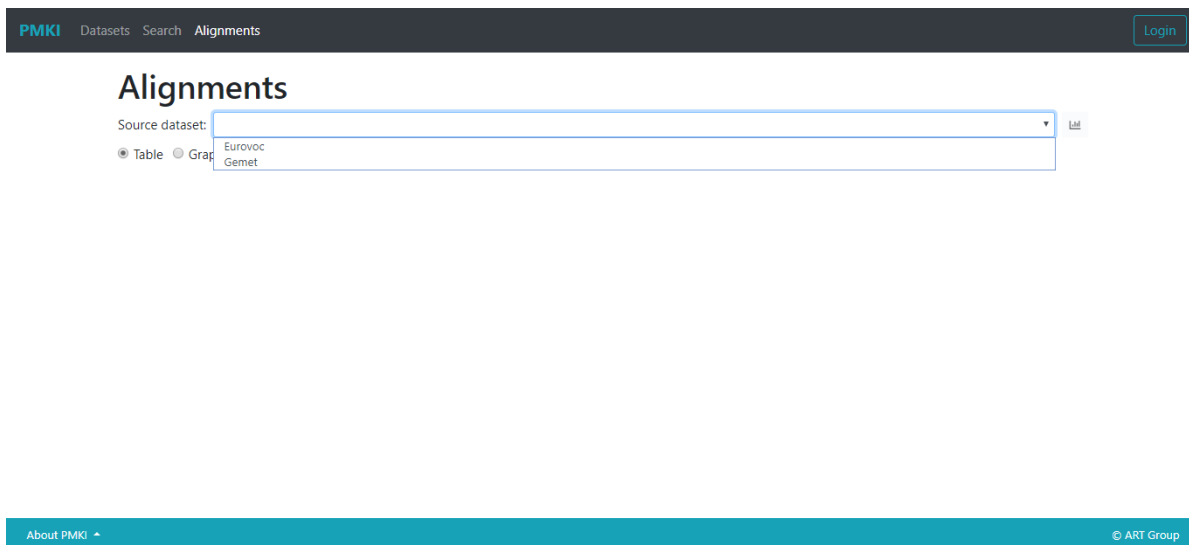
[Edit language list](#)

Currently it is possible to restrict the results to the open projects, or include also the closed ones (note that the results of closed projects cannot be inspected, they will be just shown in the results list). Another available filter is applicable to the *languages*. A set of (customizable) languages can be selected in order to search matches only in literal values that are expressed in that language.

4.4 ALIGNMENTS

4.4.1 INTRODUCTION

The *Alignments* page can be accessed from the top navigation bar. This page allows to inspect the mappings defined between datasets, namely the semantic relations that link resources belonging to a source dataset to resources belonging to a target dataset.



A combobox on the top of the page allows to select the source dataset among the open projects hosted in PMKI. Then, the two radio button below allows to chose between two different way for visualizing the alignments: the *table* and the *graph* view.

4.4.2 TABLE VIEW

The alignments table view, as suggested by the name, shows the datasets linked to the source one in a two-columns table. For each target dataset is specified the amount of alignment relations defined. Note that in this list the Eurovoc dataset is the only one with a "label" since its metadata are registered in PMKI. The other target datasets are unknown, probably they are not hosted in PMKI or the metadata of the URI listed are not found in the metadata registry so the system cannot resolve the name of the dataset.

PMKI Datasets Search Alignments Login

Alignments

Source dataset: Gemet

Table Graph

Dataset	Alignments
http://data.uba.de/um/	3483
http://dbpedia.org/resource/	3006
http://aims.fao.org/aos/agrovoc/	1199
http://rdfsdata.eionet.europa.eu/eea/determinants/	32
Eurovoc	1938

About PMKI © ART Group

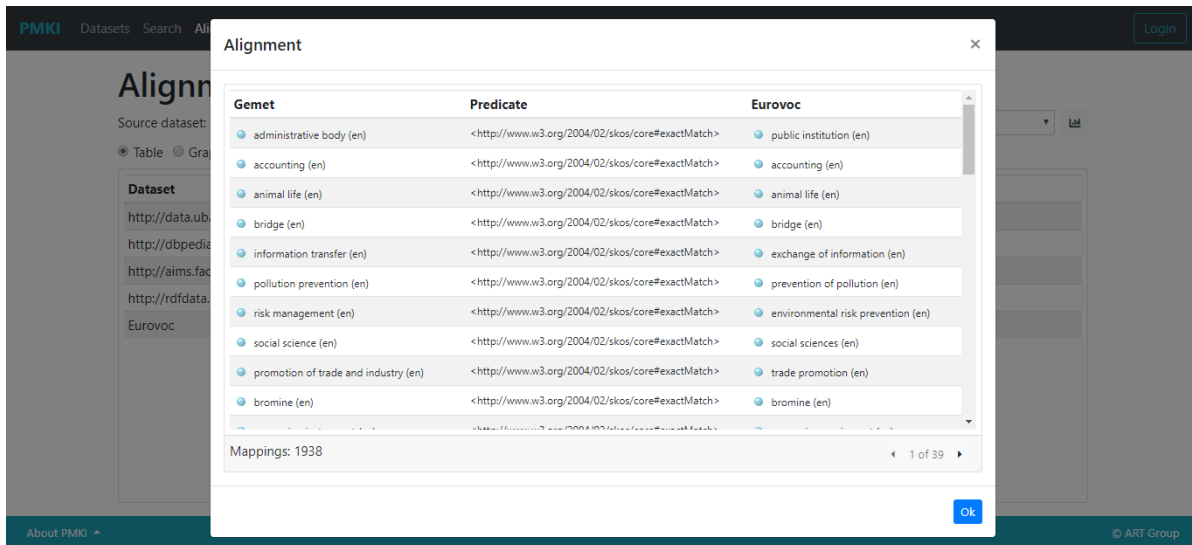
When a source dataset is selected, the system checks if the needed metadata of the dataset are available. In negative case it propose the user to generate them.

Missing profile ×

Unable to find metadata about the project 'Gemet' in the MetadataRegistry. Do you want to profile the project? (required for the alignment feature)

Yes No

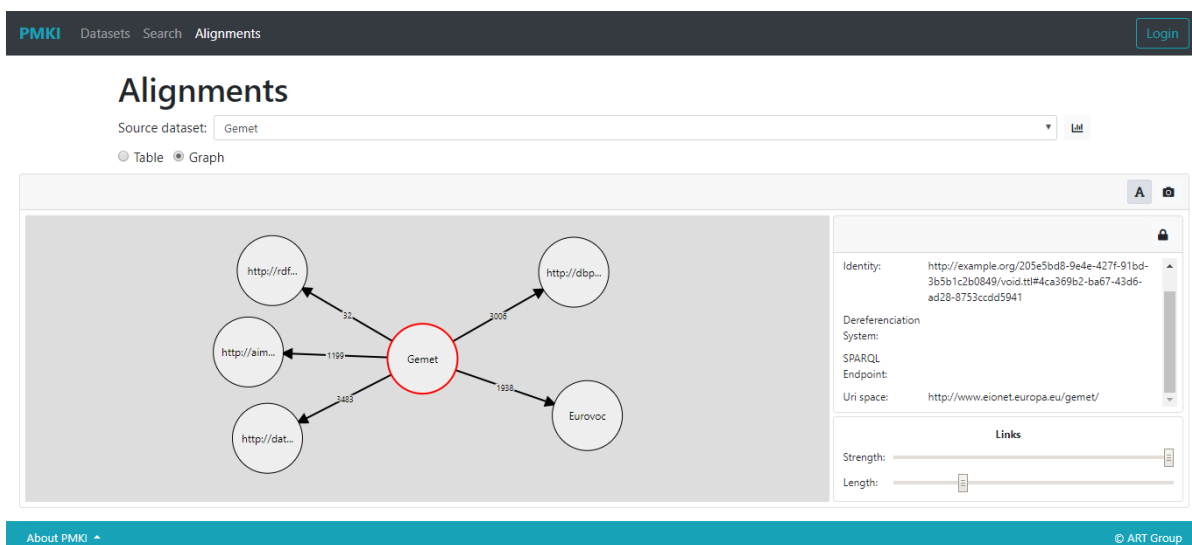
A specific alignment set can be inspected simply by clicking on the desired row. A dialog shows the effective triples subject-predicate-object where the subject belongs to the source dataset and the object to the target. Clicking on a resource is then possible to inspect its ResourceView in a modal dialog (target resources might not be inspected if they belong to a dataset not hosted in PMKI or if the metadata does not specify any valid SPARQL endpoint).



4.4.3 GRAPH VIEW

An alternative visualization mode is the Graph view. This view (very similar to the one in the [Data](#) view) allows the interactive visualization of the alignments with a force-directed graph (based on the [d3.js](#) library). The user can interact with the graph by selecting elements, exploring nodes and customizing the visualization.

The graph view is composed by a single panel splitted in two main sections: the left one which shows the graph and the right one containing a panel that shows a brief description of the selected node. At the bottom of the right panel there are also two sliders that allow to change the strength and the length of the links.



Each node of the graph represents a dataset while the edges represent the linksets that go from the source to the target node/dataset.

Every element of the graph, be it a node or an edge, when clicked is highlighted with a red border. Then the resource represented by the node is briefly described in the right panel. In case the selected element is a node, so a dataset, the panel contains its metadata, otherwise in case it is an edge, so a linksets, the panel indicates the source and the target datasets, the mappings count (also visible at the middle of the edge) and also a link to inspect the mappings list (the same described in the table view). On the top on this panel, a lock button allows to lock/unlock the node position and so avoid that the node moves around the graph "pushed" by the graph forces. The alignments of a dataset can be expanded or collapsed with a double-click on the node.

End of Document